

Sistema para evaluar la calidad de un producto o servicio Software

Daniel Esteban Cano López

**Escuela de Ingeniería
Departamento de Ingeniería de Sistemas
Universidad EAFIT**

Medellín

2012

Sistema para evaluar la calidad de un producto o servicio Software

Daniel Esteban Cano López

**Trabajo de grado presentado como requisito para optar al título de Ingeniero
de Sistemas**

Asesor metodológico y temático

Rafael David Rincón Bermúdez

Profesor Departamento de Informática y Sistemas

Universidad EAFIT

Escuela de Ingeniería

Departamento de Ingeniería de Sistemas

Universidad EAFIT

MEDELLÍN

2012

Nota de Aceptación:

Presidente del jurado

Jurado

Jurado

AGRADECIMIENTOS

Mi gratitud, principalmente está dirigida a Dios, por haberme dado la existencia y por darme fuerzas para seguir siempre hacia adelante, demostrándome su infinita bondad y amor.

Quiero agradecer y dedicar este trabajo a mi esposa Dailys Mejía y a mi hijo Daniel Camilo Cano Mejía, quienes han sido el mayor apoyo y motivación en la conclusión de este proyecto.

Siento una gran satisfacción por el resultado obtenido en este proyecto, gracias a la asesoría incondicional del Profesor Rafael David Rincón, quien me brindó siempre su orientación con profesionalismo.

Agradezco también a mis padres, por su confianza, por darme ejemplos dignos de superación y por haber fomentado en mí el deseo de salir adelante y el anhelo de triunfo en la vida.

RESUMEN

SERVIQUALITY pretende facilitar el proceso de calificación de un producto bien o un servicio de software.

Con el fin de poderse integrar con otros productos manejados en las organizaciones, SERVIQUALITY se encuentra conformado por una serie de portlets, los cuales podrán ser instalados en un portal (Integrador de herramientas Web).

TABLA DE CONTENIDO

1.	PRELIMINARES.....	15
1.1	DEFINICIÓN DEL PROBLEMA.....	15
1.2	OBJETIVO	16
1.2.1	Objetivo General.....	16
1.2.2	Objetivos Específicos	16
2.	MARCO TEÓRICO.....	17
2.1	PRÓLOGO	17
2.2	MODELOS DE CALIDAD	17
2.3	REQUISITOS	22
2.3.1	Requisitos Funcionales.....	22
2.3.2	Requisitos No Funcionales	23
2.4	SELECCIÓN DE LA TECNOLOGÍA.....	25
2.5	ARQUITECTURA DE SISTEMAS	30
2.5.1	Actores	31
2.5.2	Portlet maestroNodosPQPortlet.....	33
2.5.3	CU-01 Administración de tipos de nodos.....	33
2.5.4	CU-02 Administración Nodos.....	36
2.5.5	Portlet maestroPlantillasPQPorlet.....	39
2.5.6	CU-07 Administración de plantillas	40
2.5.7	Portlet maestroProyectosPQPortlet	44
2.5.8	Portlet calificacionPQPortlet	49
2.5.9	Casos de uso vs Requisitos funcionales	53
2.6	METODOLOGÍA DE TRABAJO	56
2.7	DESPLIGUE DE LA APLICACIÓN.....	59
2.8	AMBIENTE DE DESARROLLO – PRUEBAS.....	61
2.9	ARQUITECTURA DE COMPONENTES	64
2.10	MODELO DE CLASES – ENTIDADES	66
2.11	MODELO DE DATOS	71

CONCLUSIONES Y OPORTUNIDADES DE MEJORA	87
BIBLIOGRAFÍA	89
ANEXO A – TÉRMINOS Y DEFINICIONES.....	91
ANEXO B – PRUEBA PILOTO.....	98
ANEXO C – PRUEBAS NO FUNCIONALES	101

LISTA DE TABLAS

Tabla 1. Tablas vs Clases entidad.	66
Tabla 2. Modelo de clases - Entidades. Parte 1.	67
Tabla 3. Modelo de clases - Entidades. Parte 2.	68
Tabla 4. Modelo de clases - Entidades. Parte 3.	69
Tabla 5. TSQ_TIPO_NODO – Columnas.	71
Tabla 6. TSQ_TIPO_NODO – Constraints.	71
Tabla 7. TSQ_TIPO_NODO - Relationships.	72
Tabla 8. TSQ_NODO - Columnas.	72
Tabla 9. TSQ_NODO – Constraints	72
Tabla 10. TSQ_NODO - Relationships.	73
Tabla 11. TSQ_PLANTILLA_NODO – Columnas.	73
Tabla 12. TSQ_PLANTILLA_NODO - Constraints.	74
Tabla 13. TSQ_PLANTILLA_NODO - Relationships.	75
Tabla 14. TSQ_PLANTILLA – Columnas	75
Tabla 15. TSQ_PLANTILLA - Constraints.	75
Tabla 16. TSQ_PLANTILLA - Relationships.	76
Tabla 17. TSQ_PROYECTO_NODO - Columnas.	78
Tabla 18. TSQ_PROYECTO_NODO – Constraints	78
Tabla 19. TSQ_PROYECTO_NODO - Relationships.	79
Tabla 20. TSQ_PROYECTO - Columnas.	80
Tabla 21. TSQ_PROYECTO - Constraints.	80
Tabla 22. TSQ_PROYECTO - Relationships.	80
Tabla 23. TSQ_PROYECTO_USER - Columnas.	82
Tabla 24. TSQ_PROYECTO_USER - Constraints.	82
Tabla 25. TSQ_PROYECTO_USER - Relationships.	83
Tabla 26. TSQ_CAL_PROYECTO_NODO - Columnas.	84
Tabla 27. TSQ_CAL_PROYECTO_NODO - Constraints.	84
Tabla 28. TSQ_CAL_PROYECTO_NODO - Relationships.	85

Tabla 29. Prueba piloto.	98
-------------------------------	----

LISTA DE FIGURAS

Figura 1. Modelo de calidad de McCall.	18
Figura 2. Modelo de Boehm para Clasificar los Criterios de Calidad.	19
Figura 3. Modelo de Calidad para Calidad Externa e Interna (Norma ISO/IEC 9126-1:2001).	20
Figura 5. Modelo de Calidad para Calidad Externa e Interna (norma SQuaRE - ISO/IEC 25010).	21
Figura 6. Requisitos funcionales.	22
Figura 7. Requisitos no funcionales.	23
Figura 8. Modelo de casos de uso.	30
Figura 9. Grupos de perfiles.	31
Figura 10. Actores externos.	31
Figura 11. Actores internos.	32
Figura 12. Funcionalidades que provee el portlet maestroNodosPQPortlet.	33
Figura 13. Actividades administrar tipos de nodos.	34
Figura 14. Lista de tipos de nodos y filtro de consulta.	34
Figura 15. Formulario crear/editar tipo de nodo.	35
Figura 16. Actividades administrar nodos.	37
Figura 17. Lista nodos registrados y filtro de consulta.	37
Figura 18. Formulario crear/editar nodo.	38
Figura 19. maestroPlantillasPQPortlet.	39
Figura 20. Administración de plantillas.	40
Figura 21. Lista de plantillas y filtro de consulta.	41
Figura 22. Adicionar/Editar plantilla.	42
Figura 23. Configurar plantilla.	43
Figura 24. Adicionar/Eliminar nodo.	44
Figura 25. Casos de usos.	45
Figura 26. Lista de proyectos.	46
Figura 27. Adicionar, editar, calificar proyecto.	47

Figura 28. Configurar nodos del modelo. Adicionar conclusión al resultado obtenido por nodo.	48
Figura 29. Calificadores asignados al proyecto.	49
Figura 30. Adicionar/Editar usuario.	49
Figura 31. Casos de uso que implementa el portlet calificacionPQPortlet.	50
Figura 32. Actividades calificar modelo.	50
Figura 33. Carga de proyectos.	51
Figura 34. Selección del atributo.	52
Figura 35. Casos de uso vs requisitos funcionales.	55
Figura 36. Metodología de trabajo.	57
Figura 37. Diagrama de despliegue.	60
Figura 38. Diagrama de despliegue.	63
Figura 43. Arquitectura de componentes.	65
Figura 44. Modelo de clases - Entidades.	70
Figura 45. Modelo de datos. Plantilla.	76
Figura 46. Modelo de datos. Proyectos.	81
Figura 47. Modelo de datos. Calificación.	86
Figura 48. Árbol.	93
Figura 49. Árbol.	94
Figura 50. Proyectos finalizados satisfactoriamente (5 de 12).	100
Figura 39. Escenario prueba 1.	101
Figura 40. Tiempos de respuesta 10 usuarios.	102
Figura 41. Configuración JMeter.	103
Figura 42. Resultados JMeter.	103

INTRODUCCIÓN

El uso de herramientas tecnológicas permite a las empresas obtener ventajas competitivas como consecuencia de la agilización de procesos, y un notable rendimiento general en la organización. El incremento en la complejidad de productos software se evidencia en usuarios cada vez más exigentes en cuanto a calidad se refiere.

Esta exigencia y la complejidad del software ameritan que sea estrictamente analizado bajo estándares de calidad, garantizando un producto que cumpla con los requerimientos de los usuarios y del negocio.

“La calidad de un producto de software se debe evaluar usando un modelo definido. El modelo de calidad debe ser utilizado al fijar las metas de la calidad para los productos de software y los productos intermedios. Dado que es imposible medir la calidad de un producto de software de forma directa, éste se descompone jerárquicamente en un modelo de calidad constituido por características y sub-características.”¹

Con este trabajo se pone a disposición de la comunidad informática una herramienta que permita implementar y adaptar los modelos existentes para la evaluación de la calidad de productos software, concebidos a manera de un diagrama de árbol (“Divide y conquistarás”).

Luego de hacer un análisis de la compañía Repuestos Pedregal SA, se identificó que se requiere contar con un CRM, una herramienta que apoye la gestión de las

¹ GONZÁLEZ JARDON, Carlos. La gestión del Proceso Software. Primera Parte. Septiembre 09. Año 2006.

relaciones con los clientes, las ventas y el marketing. La compañía lanzó una licitación en la que tres empresas de TI, enviaron sus propuestas.

La persona encargada del proceso de adquisición de la herramienta (Gerente de Tecnología) evaluó las tres soluciones suministradas con el fin de optar por la que más se ajustara a los requerimientos de la compañía.

El Gerente de Tecnología decidió hacer uso del “Modelo de Calidad para Calidad Externa e Interna (norma SQuaRE - ISO/IEC 25010)”, para realizar dicha evaluación. Con el propósito de facilitar este proceso, el Gerente de Tecnología hace uso de SERVIQUALITY y, estructura el proyecto indicando el nivel de importancia a ser asignado.

El resultado obtenido para las tres soluciones propuestas por las compañías licitantes, fue:

- Producto compañía 1: 3/10.
- Producto compañía 2: 7/10.
- Producto compañía 3: 9/10.

Teniendo en cuenta los resultados obtenidos luego de efectuar este proceso, el Gerente de Tecnología le indica al Gerente General que a pesar de que el producto de la compañía 1 es el más económico de todos, se debe optar por el producto de la compañía 3, por los siguientes motivos:

- Suple todas las funcionalidades requeridas.
- Suple los requisitos no funcionales detectados.
- El precio se encuentra dentro del presupuesto.

El Gerente General, muy amigo de dueño del producto que ofrece la compañía 1, le pregunta a su Gerente de Tecnología ¿Por qué no seleccionamos el producto que ofrece la compañía 1?

El Gerente de Tecnología responde:

- Suple sólo un 70% las funcionalidades requeridas.
- La construcción de las demás funcionalidades posiblemente impacte de forma negativa las ya construidas, por lo que habría que ajustarlas. El tiempo de entrega para el resto de las funcionalidades sería de 6 meses y su implementación será de \$"X".
- Si tenemos presente la inversión inicial, el costo de desarrollar las demás funcionalidades, el tiempo requerido para tener todo funcionando a cabalidad, los ingresos que dejaría de generar la compañía en este lapso de tiempo, entre otros aspectos, encontramos que para nuestra empresa le resulta más costoso adquirir el producto que ofrece la compañía 1 con relación a lo ofrecido por la compañía 3.

¿Qué cree usted que hubiese sucedido al no realizar este estudio y el Gerente General decide que debe ser elegida la opción 1, tomando como argumento de decisión la buena relación que se tiene con el propietario de dicha compañía?

1. PRELIMINARES

1.1 DEFINICIÓN DEL PROBLEMA

Evaluar la calidad de un servicio o producto de software presenta una serie de dificultades, entre las cuales se encuentran:

- Subjetividad en el proceso de evaluación.
- Falta de herramientas que faciliten el proceso de evaluación y posibiliten el análisis.
- Es un proceso dispendioso y en general no se cuenta con una estructura formal y organizada para dicho proceso.

La construcción de SERVIQUALITY, busca entre otros aspectos:

1. Proporcionar una herramienta a las empresas que quieren invertir en TI y no cuentan con los mecanismos necesarios para elegir la mejor solución en términos de costo, funcionalidad, disponibilidad, portabilidad, amigabilidad, entre muchos otros aspectos.
2. Poner a disposición de las compañías de TI una herramienta que permita validar la calidad de sus productos o servicios, con el fin de apoyar el ciclo de mejora continua en la organización. De esta manera, tanto el cliente como el proveedor podrán analizar de una forma más objetiva si el resultado obtenido con el producto o servicio cumple con lo pactado.

1.2 OBJETIVO

1.2.1 Objetivo General

Construir una herramienta de software que permita calificar un producto o servicio de software con base en estándares internacionales de calidad.

1.2.2 Objetivos Específicos

- Definir los requisitos funcionales del sistema en base a la estructura que manejan los estándares internacionales relacionados con la evaluación de productos y servicios de software.
- Establecer una arquitectura que posibilite la construcción de portlets para el portal Liferay en base a buenas prácticas.
- Construir la aplicación con base a las prácticas propuestas por la Ingeniería de Software.
- Garantizar el funcionamiento de SERVIQUALITY con pruebas pilotos, validaciones con expertos, pruebas unitarias y pruebas de rendimiento.

2. MARCO TEÓRICO

2.1 PRÓLOGO

En un mundo globalizado, las empresas se ven enfrentadas a competir para su sostenibilidad, por tal motivo la calidad es un factor clave y diferenciador, además de aumentar la satisfacción de los usuarios y disminuir costos, entre otros aspectos relevantes.

El software posee un papel primordial puesto que logra agilizar procesos, trayendo consigo la optimización de recursos. Por esta razón la evaluación de un producto software es conveniente para conocer el nivel de cumplimiento de lo requerido por el producto software.

2.2 MODELOS DE CALIDAD

Previo a este trabajo se analizó la estructura que manejan los siguientes modelos de calidad.

- Modelo de calidad de McCall.
- Modelo de Boehm para Clasificar los Criterios de Calidad.
- Modelo de Calidad para Calidad Externa e Interna (Norma ISO/IEC 9126-1:2001).
- Modelo de Calidad para Calidad Externa e Interna (norma SQuaRE - ISO/IEC 25010).

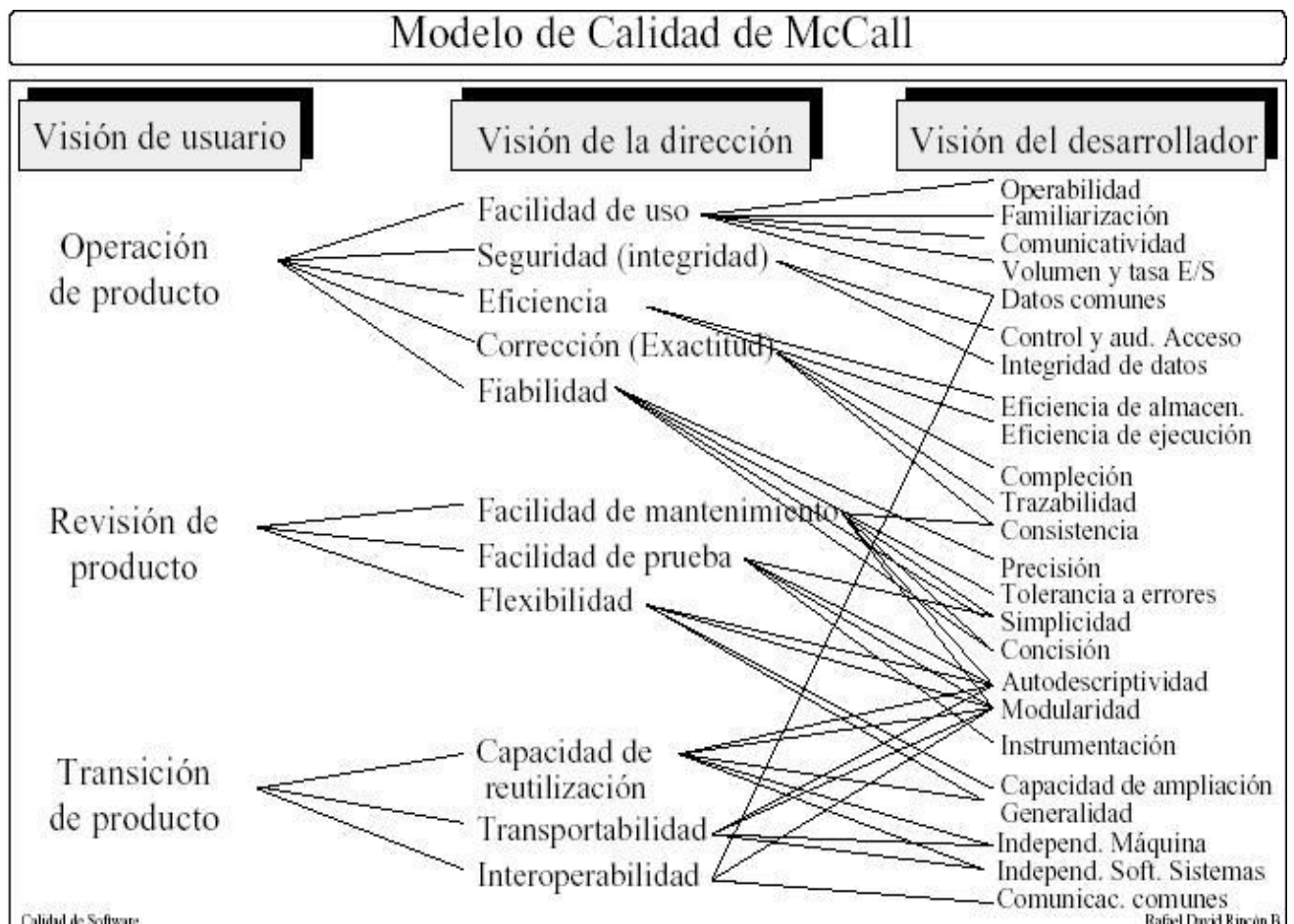


Figura 1. Modelo de calidad de McCall.²

El modelo de McCall define tres ejes o puntos de vista (Operación de producto, Revisión de producto y transición de producto) partir de los cuales se puede identificar la calidad de un producto.

Cada uno de estos ejes se encuentra basado en un conjunto de factores (Visión de la dirección) los cuales a su vez se basan en un conjunto de criterios (Visión del desarrollador).³

² **Piedrahita Mesa, Sebastián. 2007.** *Construcción de una herramienta para evaluar la calidad de un producto software.* Medellín : Universidad EAFIT, 2007.

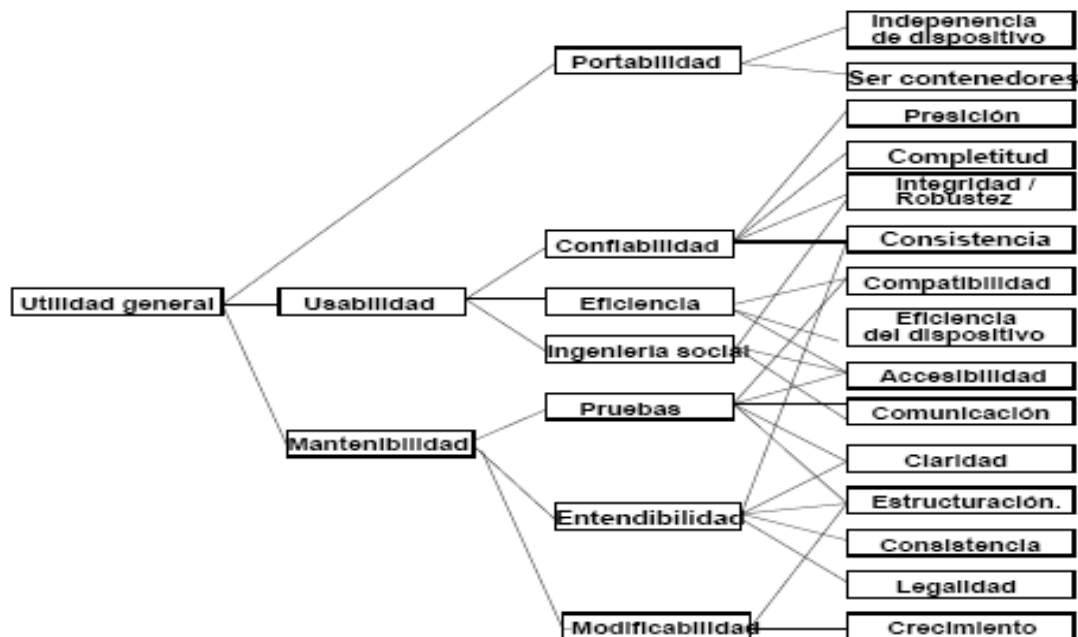


Figura 2. Modelo de Boehm para Clasificar los Criterios de Calidad.⁴

Modelo de calidad propuesto por Barry Boehm en 1978. Este modelo maneja tres niveles, características de alto nivel, características de nivel intermedio y características primitivas.

Con las características de alto nivel se definen y validan las características generales de uso; las características de nivel intermedio representan los factores de calidad de Boehm y las características primitivas hacen referencia a una o varias métricas de calidad.⁵

³ Paz, Angel Cervera. Monografias. [En línea] [Citado el: 14 de 04 de 2012.] <http://www.monografias.com/trabajos5/call/call.shtml>.

⁴ Piedrahita Mesa, Sebastián. 2007. *Construcción de una herramienta para evaluar la calidad de un producto software*. Medellín : Universidad EAFIT, 2007.

⁵ Fillottrani, Pablo R. Departamento de ciencias é ingeniería de la computación. Universidad nacional del sur. Buenos Aires. Argentina. [En línea] [Citado el: 14 de 04 de 2012.] <http://www.cs.uns.edu.ar/~prf/teaching/SQ07/clase6.pdf>

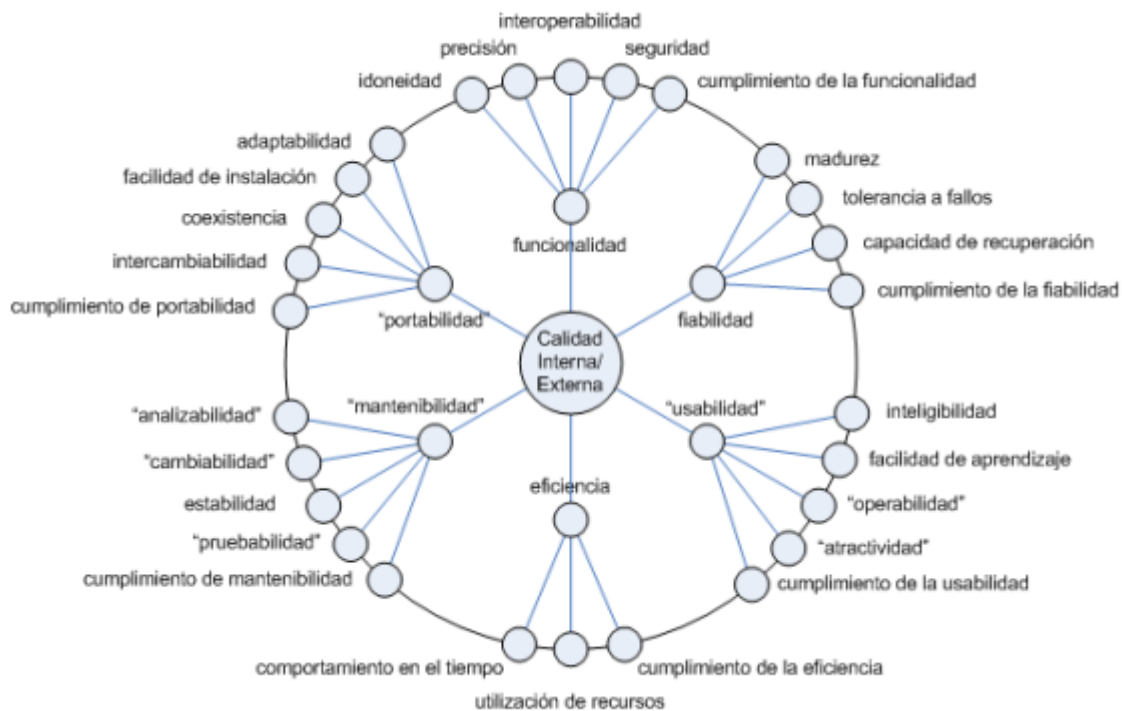


Figura 3. Modelo de Calidad para Calidad Externa e Interna (Norma ISO/IEC 9126-1:2001).⁶

El modelo propuesto por el estándar ISO 9126, propone un modelo de calidad que se divide en dos vistas interior y exterior las cuales están compuestas por seis características (Funcionalidad, Fiabilidad, Usabilidad, Eficiencia, Mantenibilidad, Portabilidad).

Cada característica está compuesta por una serie de subcaracterísticas, cada subcaracterísticas comprende un conjunto de atributos los cuales hacen uso de diferentes métricas para calcular su valor.

⁶ **ISO25000.** ISO25000. [En línea] [Citado el: 17 de Abril de 2012.] <http://iso25000.com/index.php/iso-iec-9126.html>.

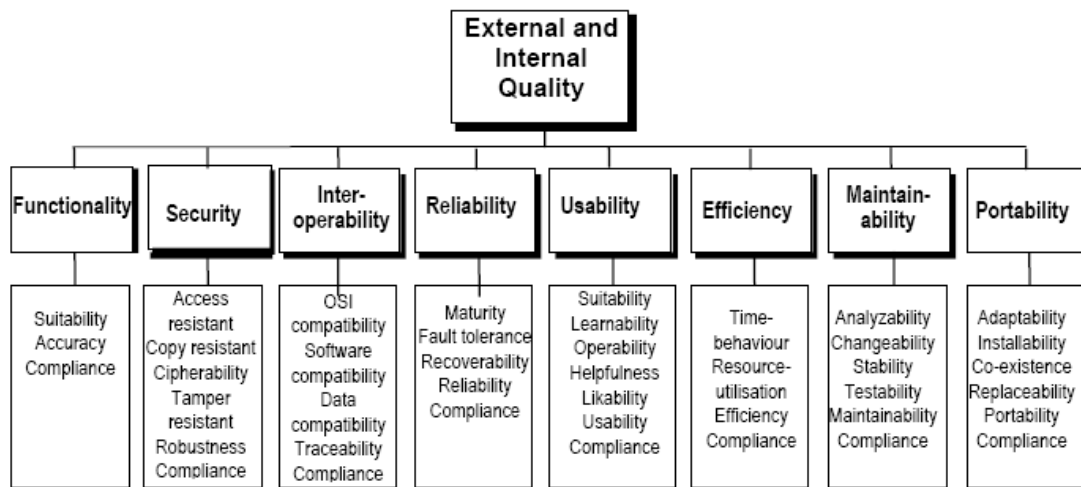


Figura 4. Modelo de Calidad para Calidad Externa e Interna (norma SQuaRE - ISO/IEC 25010).⁷

El modelo de Calidad para Calidad Externa e Interna (norma SQuaRE 25010), es una evolución de la norma propuesta por la ISO 9126. En este modelo se adiciona la vista en uso la cual está compuesta por las siguientes características: Efectividad, productividad, seguridad y satisfacción.

Como se puede constatar en los principales modelos de calidad, anteriormente mencionados, estos cuentan con una estructura de diagrama de árbol⁸ para facilitar el proceso de evaluación.

Con el fin de brindar una solución estándar, se identificó que SERVIQUALITY debería contar con la posibilidad de administrar estructuras de datos en forma de árbol, las cuales se convertirían en modelos de calificación de productos o servicios software. Con esto se busca que los usuarios puedan definir sus propios modelos de calificación o adaptar los modelos existentes.

⁷ **International Standards for Business, Government and Society. 2007. Software engineering - Software product Quality Requirements and Evaluation (SQuare) - quality model.** s.l. : ISO and IEC, 2007. ISO/IEC 25010.

⁸ Ver Árbol – Nodo – Hoja. Anexo A – Términos y definiciones.

2.3 REQUISITOS

A continuación se exponen los requisitos funcionales y no funcionales que permiten determinar si el sistema cumple con la calidad esperada.

2.3.1 Requisitos Funcionales

Los requisitos que se definen a continuación determinaran el funcionamiento interno de la aplicación.

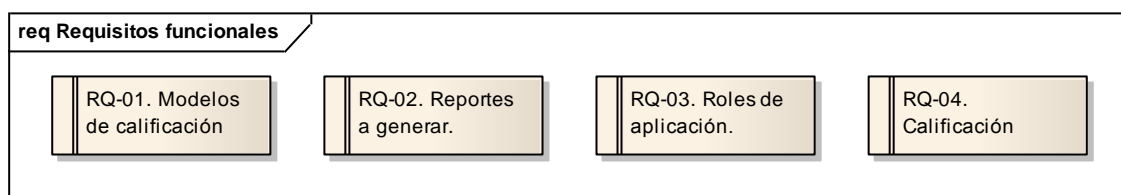


Figura 5. Requisitos funcionales.⁹

Los requisitos funcionales detectados son:

- RQ-01. Modelos de calificación. El sistema debe permitir implementar diferentes modelos de calificación que sigan una estructura de árbol; permitiéndole al usuario configurar los siguientes aspectos:
 - Un nivel de importancia en cada nodo.
 - Valor esperado en cada nodo.
- RQ-02. Reportes a generar. A partir de una grafica de radar el sistema deberá posibilitar comparar los valores esperados y obtenidos luego de realizar el proceso de evaluación.
- RQ-03. Roles de aplicación. La aplicación debe manejar los siguientes roles:

⁹ Elaboración propia

- Calificador. Persona encargada de realizar la evaluación. Un usuario calificador puede hacer parte del de todos o algunos equipos de evaluación de productos de software, con el fin de garantizar la confidencialidad de los datos, el usuario calificador solamente podrá acceder a la información de los proyectos donde es participe.
- Líder Gestor. Persona encargada de definir y configurar el modelo a emplear para realizar el proceso de evaluación.
- RQ-04. Calificación. El sistema debe permitir la calificación de un producto Software haciendo uso de un modelo de calificación que siga la estructura de árbol. Previo a la realización de la evaluación del producto software el Líder Gestor ingresa al sistema los valores esperados y el nivel de importancia que se otorgara a cada nodo del modelo (En el modelo SQuaRE 25010 serian las características, subcaracterísticas o atributos).

2.3.2 Requisitos No Funcionales

Adicional al cumplimiento de los comportamientos esperados, el sistema debe cumplir con los siguientes requisitos no funcionales, los cuales al igual que los requisitos funcionales, serán criterios claves para determinar si el sistema presenta los requerimientos esperados.

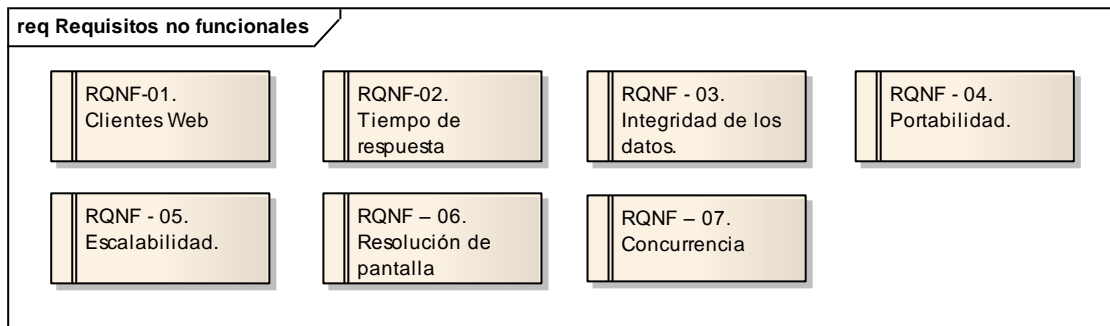


Figura 6. Requisitos no funcionales.¹⁰

¹⁰ Elaboración propia.

Se han identificado los siguientes requisitos no funcionales que debe suplir el sistema:

- RQNF-01. Clientes Web. El sistema debe poderse visualizar correctamente en los siguientes navegadores:
 - Mozilla Firefox 4 o superiores.
 - Internet Explorer 8 o superiores.
- RQNF-02. Tiempo de respuesta. El sistema no debe tardar más de cinco segundos en mostrar los resultados de una búsqueda. Si se supera este plazo, el sistema detiene la búsqueda y muestra los resultados encontrados.
- RQNF - 03. Integridad de los datos.
 - El sistema no puede permitir hacer inyección de código SQL por medio de los formularios que maneja.
 - El sistema debe validar los tipos de datos antes de proceder al almacenamiento.
 - Validaciones en el servidor. La solución a crear va ser cliente servidor, donde el cliente es el navegador web permitido para acceder a la aplicación y el servidor es la aplicación SERVIQUALITY. Todas las validaciones que se deban efectuar sobre los datos diligenciados por los usuarios, calificador o gestor, se deberán realizar en el servidor.
- RQNF - 04. Portabilidad. El sistema no debe depender de un motor de base de datos relacional específico.
- RQNF - 05. Escalabilidad. El sistema debe permitir a futuro adicionar nuevas funcionalidades como:
 - Reportes estadísticos.
 - Integración con otros sistemas para la captura de datos a emplear en cada métrica.
 - Configuración de las formulas de cada métrica a emplear en el modelo de evaluación.

- RQNF – 06. Resolución de pantalla. Se debe garantizar la correcta visualización de la aplicación en monitores que manejen una resolución mínima de 1024 x 768.
- RQNF – 07. Concurrencia. El sistema de soportar 50 usuarios y 10 de manera concurrente.

2.4 SELECCIÓN DE LA TECNOLOGÍA

SERVIQUALITY, debe permitir perfilar el acceso a la información (Requisito funcional RQ-03), para realizar esta labor se han identificado las siguientes alternativas.

- Construir un sistema de seguridad que pueda ser administrable.
- Adquirir un sistema de seguridad para ser utilizado en el proyecto.
- Diseñar la solución haciendo uso de portlets¹¹ los cuales serán instalados en un portal que permita perfilar el acceso a la información.

Haciendo un análisis previo se determino optar por que la solución sea construida haciendo uso de diferentes portlets. Las razones para determinar este tipo de solución fueron:

- Construir un sistema de seguridad administrable implicaría mayor inversión de trabajo incrementando la duración del proyecto y en última instancia saliéndose del alcance del mismo.
- Por costos del proyecto no se puede adquirir el sistema de seguridad.

¹¹ Ver Portlet en Anexo A – Términos y definiciones.

- Un portal web, adicional al sistema de seguridad que provee, pone a disposición otro conjunto de herramientas, las cuales pueden complementar el proceso de calificación de un producto software.
- Un portal web, posibilita la integración de SERVIQUALITY, con otros sistemas, proporcionando un único punto de entrada a las diferentes aplicaciones que se tenga en la organización que va hacer uso del aplicativo.
- Un portlet suple una funcionalidad específica, por lo que si en algún momento se requieren funcionalidades adicionales se pueden crear nuevos portlets, de esta manera se garantiza la escalabilidad de la solución.
- En caso de presentarse una dificultad en alguna funcionalidad específica o requerir actualizar dicha funcionalidad, no es necesario desinstalar toda la aplicación. Gracias a que cada portlet suple una funcionalidad específica, solo se requiere realizar la actualización del portlet, por lo tanto el impacto sería menor comparado con la actualización de toda la aplicación.
- Un portlet puede ser instalado en cualquier portal que soporte el estándar jsr-168 y jsr-286.¹²

En el mercado existen varios portales que pueden ser empleados para la construcción de la solución final. Realizando un estudio previo se identificaron los siguientes sitios donde expertos en el tema brindan sus diferentes puntos de vista.

- <http://java-junction.blogspot.com/2008/12/jboss-portal-vs-liferay-portal.html>
[Citado el: 15 de 04 de 2012.]
- http://www.manageability.org/blog/stuff/open_source_portal_servers_in_java
[Citado el: 15 de 04 de 2012.]

¹² **Oracle.** [En línea] [Citado el: 15 de 04 de 2012.]
<http://developers.sun.com/portalserver/reference/techart/jsr168/>
<http://www.jcp.org/en/jsr/detail?id=286> y

- [http://dw.riik.ee/@api/deki/files/232/=Liferay_Overview - Intalio Conf.pdf](http://dw.riik.ee/@api/deki/files/232/=Liferay_Overview_-_Intalio_Conf.pdf)
[Citado el: 15 de 04 de 2012.]
- Ventajas de usar Liferay. <http://www.liferay.com/products/liferay-portal/benefits/user-friendly> [Citado el: 15 de 04 de 2012.]
- Casos de éxito usando Liferay. <http://www.liferay.com/products/liferay-portal/stories> [Citado el: 15 de 04 de 2012.]
- “En mi empresa hemos apostado firmemente por esta tecnología, y hemos afrontado con éxito algunos portales basados en Liferay. Hemos integrado Liferay con Alfresco (gestor documental). Nuestro objetivo: ser referencia de Liferay en España, no solo para la empresa privada, sino para la Administración Pública.”
<http://caraballomaestre.blogspot.com/2009/03/portlets-con-liferay.html>
[Citado el: 15 de 04 de 2012.]
- Liferay. “La configuración del mismo, por parte de los usuarios resulta cada vez más sencilla.” <http://enalablog.com/liferay-portal> [Citado el: 15 de 04 de 2012.]
- <http://holisticsecurity.wordpress.com/2011/06/14/que-solucion-portal-deberia-usar/> [Citado el: 15 de 04 de 2012.].

De estas opiniones y estudios referente al tema destaco los presentados por Roger Carhuatocto. Los cuales se encuentran en <http://holisticsecurity.wordpress.com/2011/06/14/que-solucion-portal-deberia-usar/> [Citado el: 15 de 04 de 2012.]. A continuación adjunto el resultado del analisis realizado.

- Maneja una versión “Community”, la cual puede ser empleada de manera gratuita.
- Cuenta con una gran comunidad que constantemente contribuye a la mejoría del portal.
- Soporta el estándar JSR - 286 (Portlets 2.0). Posibilita la creación de aplicaciones que funcionan sobre el portal.
- Cuenta con un sistema propio de seguridad.
- Permite ser utilizado como CMS, con esto se busca que la organización interesada en hacer uso de SERVIQUALITY, pueda publicar información que considera pertinente.
- Permite la centralización de herramientas web empleadas en una organización o áreas de negocio, facilitando el acceso.
- Permite la publicación de documentos (PDF, Word, Excel,...).
- Provee otras herramientas como foros, blogs, cliente de correo, chat entre otras las cuales son de gran utilidad para el proceso de comunicación de las personas que van a emplear SERVIQUALITY.

Buscando en diferentes sitios, información acerca de la construcción de portlets se encontró <http://www.portletfaces.org/projects/portletfaces-bridge>, el cual cuenta con una amplia documentación y diferentes ejemplos. Es por tal motivo que los frameworks a emplear para la construcción de los portlets son:

- Portlet faces bridge. Integra icefaces con portlets.
- Icefaces. Entre otras funcionalidades, proporciona diferentes componentes web que facilitan el desarrollo y posibilitan la amigabilidad de la aplicación.
- Spring. Conjunto de utilidades que posibilitan la construcción de aplicaciones escalables, fáciles de integrar con otras tecnologías entre otras bondades con las que cuenta.

- Hibernate. Permite la independencia del motor de base de datos relacional, permitiendo contar con aplicaciones portables.

2.5 ARQUITECTURA DE SISTEMAS

A continuación se ilustran los diferentes portlets a ser construidos, que usuarios interactúan con estos y las funcionalidades que cada uno de estos implementa.

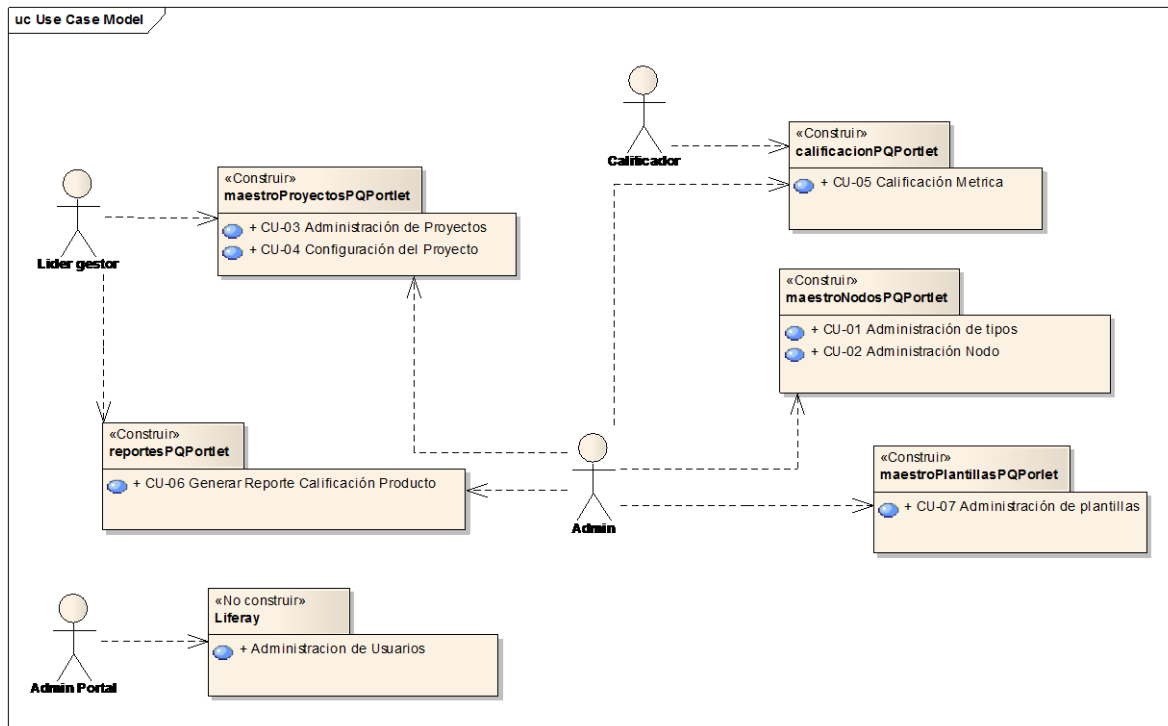


Figura 7. Modelo de casos de uso.¹⁴

¹⁴ Elaboración propia.

2.5.1 Actores

Con el fin de perfilar las funcionalidades que proveen el conjunto de portlets, se han identificado los siguientes grupos de perfiles.

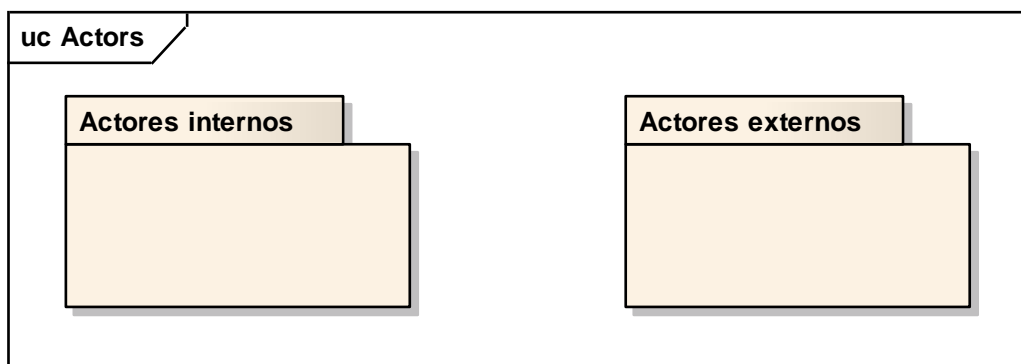


Figura 8. Grupos de perfiles.¹⁵

Actores externos

Roles que van hacer uso de los portlets y que cubren las funcionalidades necesarias para el proceso de evaluación de un producto o servicio software.

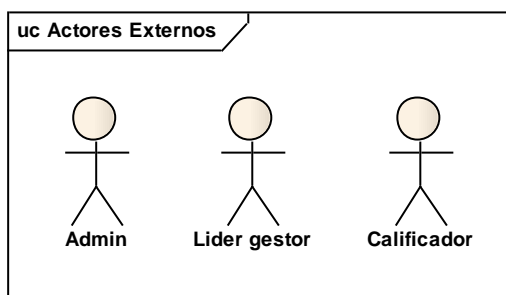


Figura 9. Actores externos.¹⁶

Admin: La persona que ejecute este rol tendrá acceso a todas las funcionalidades, aunque su principal labor será administrar los tipos de nodos, nodos, y proyectos de evaluación de productos software.

¹⁵ Elaboración propia.

¹⁶ Elaboración propia

Calificador: Como su rol lo indica, es el encargado de valorar cada métrica del modelo definido por el Líder gestor.

Líder gestor: La persona encargada de ejecutar este rol deberá definir el modelo a emplear para calificar el producto o servicio. Entre sus funciones se encuentran:

- Matricular las personas que harán parte del proceso de evaluación en el SERVIQUALITY.
- Indicar los niveles de importancia que se otorgará a cada uno de los ítems del modelo (valor esperado e importancia).

Actores internos

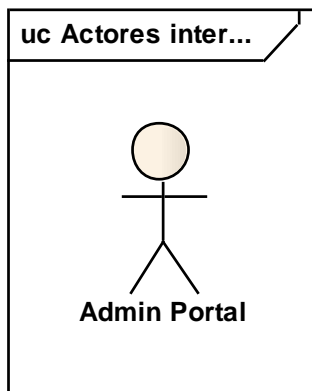


Figura 10. Actores internos.¹⁷

Admin Portal: Este es el encargado de configurar y mantener el portal.

Tareas a desempeñar:

1. Administración de los usuarios.
2. Administración de las aplicaciones en el portal.
3. Garantizar la disponibilidad del portal.

¹⁷ Elaboración propia

2.5.2 Portlet maestroNodosPQPortlet

Portlet permite la administración de nodos y tipos de nodos (Niveles que manejan los modelos) a emplear al momento de construir el modelo de evaluación.

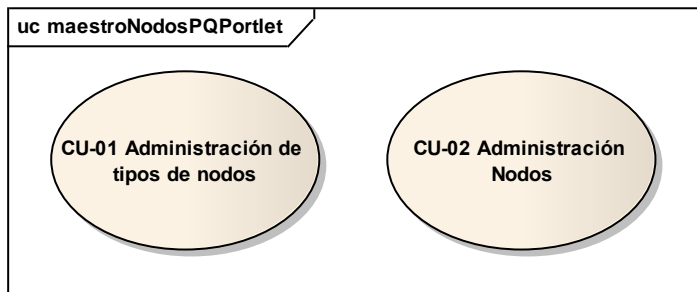


Figura 11. Funcionalidades que provee el portlet maestroNodosPQPortlet¹⁸

2.5.3 CU-01 Administración de tipos de nodos

Esta funcionalidad, permite que el usuario administre los diferentes tipos de nodos, para lo cual el usuario deberá contar con las siguientes opciones:

- Consultar todos los tipos de nodos
- Crear y editar un tipo de nodo.

¹⁸ Elaboración propia.

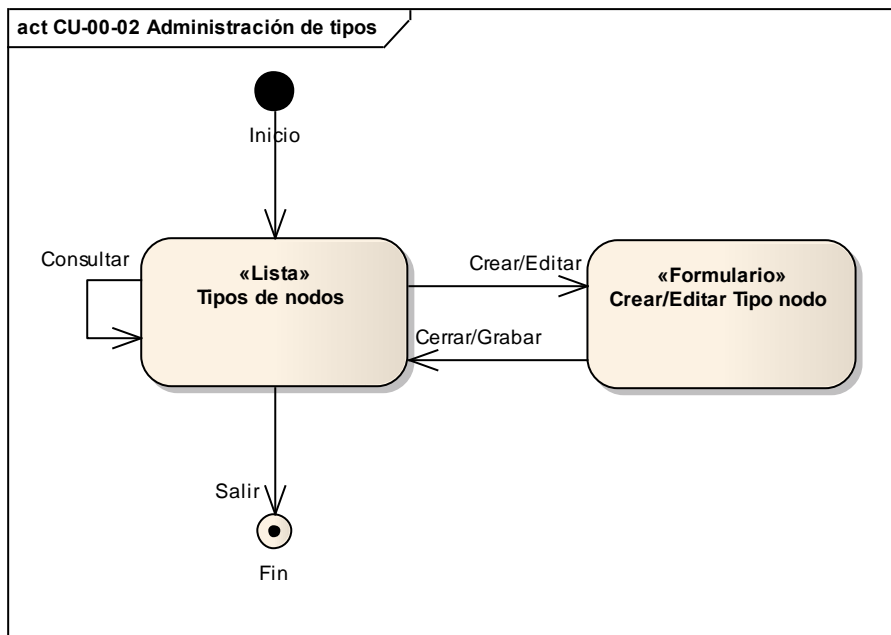


Figura 12. Actividades administrar tipos de nodos.¹⁹

Consultar Tipo de nodo

Identificador Nivel

Nombre Estado

Listado de tipos de nodos

[Adicionar un tipo de nodo](#)

Identificador	Nombre	Nivel	Estado
Características	Características	0	Activo
Subcaracterísticas	Subcaracterísticas	1	Activo
Atributos	Atributos	2	Activo

3 registros, mostrando 3 registro(s), desde 1 hasta 3. Pagina 1 / 1.

Figura 13. Lista de tipos de nodos y filtro de consulta.²⁰

Por medio de esta vista el usuario puede consultar los diferentes tipos de nodos creados en el sistema, para tal fin se pone a disposición lo siguiente:

¹⁹ Elaboración propia.

²⁰ Elaboración propia.

- Un filtro de consulta. Este permite consultar por identificador, nivel, nombre y estado; una vez se presiona el botón consultar el sistema consulta los tipos de nodos que cumplen con el criterio de selección.
- Ordenamiento por columna. Dando click sobre el nombre de la columna el usuario podrá ordenar la lista de registros por la columna de manera ascendente o descendente.
- Herramientas de navegación entre páginas o grupos de registros. El conjunto de registros obtenidos será dividido en grupos de 15 los cuales podrán ser accedidos haciendo uso de los botones que aparecen en la parte inferior de la lista.

El formulario 'Grabar Tipo de nodo' contiene los siguientes campos:

- Identificador:** Campo de texto con el valor 'Características'.
- Nivel:** Campo de texto con el valor '0'.
- Nombre:** Campo de texto con el valor 'Características'.
- Descripción:** Área de texto grande y vacía.
- Estado:** Selector de lista desplegable con el valor 'Activo'.
- Botón:** Botón 'Grabar' en la parte inferior derecha.

Figura 14. Formulario crear/editar tipo de nodo.²¹

Una vez el usuario de click sobre “Adicionar un tipo de nodo” o alguno de los registros se muestran en un formulario los siguientes campos:

²¹ Elaboración propia.

- Identificador. Código que se le va asignar al tipo de nodo.
- Nivel. Ver “Árbol – Nodo – Hoja” en “ANEXO A – TÉRMINOS Y DEFINICIONES”.
- Nombre.
- Descripción.
- Estado. Activo o Inactivo. Dado que los tipos de nodos no pueden ser eliminados debido a que esta acción generaría la eliminación de nodos, plantillas y proyectos se brinda la posibilidad que estos sean deshabilitados con el fin de no ser empleados por futuros nodos.

Por medio de este formulario el usuario podrá efectuar las siguientes acciones:

- Crear un tipo de nodo.
- Actualizar un tipo de nodo.
- Crear un tipo de nodo a partir de otro tipo de nodo. Para realizar esta acción el usuario debe seleccionar un tipo de nodo, modificar la identificación del tipo de nodo y por ultimo presionar el botón “Grabar”.

2.5.4 CU-02 Administración Nodos

Por medio de esta funcionalidad el usuario podrá crear, buscar y modificar un nodo.

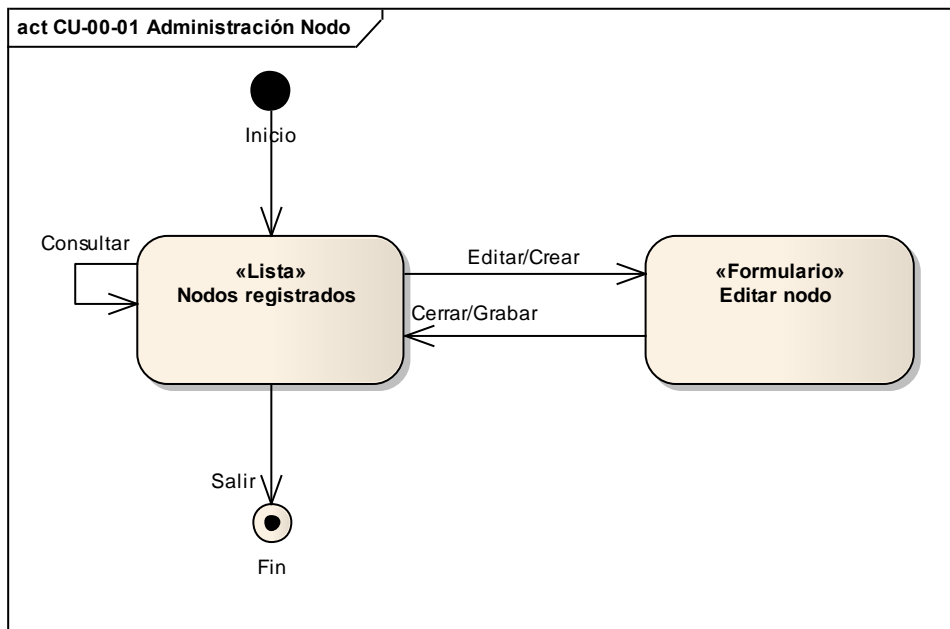


Figura 15. Actividades administrar nodos.²²

Consultar nodo

Codigo Nombre

Tipo de nodo Estado

Listado de nodos

[Adicionar un Nodo](#)

Codigo	Nombre	Tipo de nodo	Estado
SUBCAR_RESACC	Resistente al Acceso	Subcaracterísticas	Activo
SUBCAR_RESCOPIA	Resistente a la copia	Subcaracterísticas	Activo
SUBCAR_RESFALS	Resistente a la falsificación	Subcaracterísticas	Activo
SUBCAR_ROB	Robustez	Subcaracterísticas	Activo
SUBCAR_TOLFALLOS	Tolerancia a fallos	Subcaracterísticas	Activo
SUBCAR_TRAZA	Trazabilidad	Subcaracterísticas	Activo
SUBCAR_UTILREC	Utilización de recursos	Subcaracterísticas	Activo
ATTR_1	Alcance de la implementación funcional	Atributos	Activo
ATTR_2	Estabilidad (o volatilidad) de la especificación funcional	Atributos	Activo
ATTR_3	Suficiencia Funcional	Atributos	Activo

148 registros, mostrando 10 registro(s), desde 1 hasta 10. Pagina 1 / 15.

Figura 16. Lista nodos registrados y filtro de consulta.²³

Esta vista le permite al usuario consultar los nodos que han sido creados en el sistema, para realizar esta labor se le proporciona al usuario las siguientes opciones:

²² Elaboración propia.

²³ Elaboración propia.

- Filtro de consulta. El usuario podrá consultar por código, nombre, tipo de nodo (autocomplete) y estado.
- Ordenamiento por columna.
- Herramientas de navegación entre páginas o grupos de registros.

El formulario 'Grabar Nodo' tiene un título con un botón de cerrar (X roja). Los campos son:

- Código:** Campo de texto con el valor 'SUBCAR_RESACC'.
- Nombre:** Campo de texto con el valor 'Resistente al Acceso'.
- Descripción:** Área de texto con el valor 'La capacidad del software para proteger de accesos ilegales y no autorizados.'.
- Tipo de nodo:** Campo de texto con el valor 'Subcaracterísticas'.
- Estado:** Selector de lista desplegable con el valor 'Activo'.

En la esquina inferior derecha hay un botón 'Grabar'.

Figura 17. Formulario crear/editar nodo.²⁴

Una vez el usuario da click sobre “Adicionar un nodo” ó sobre alguno de los registros, el sistema le muestra un formulario con los siguientes campos:

- Código. Identificado asignado al nodo.
- Nombre.
- Descripción.

²⁴ Elaboración propia.

- Tipo de nodo. Autocomplete que permite buscar el tipo de nodo al cual será asociado el nodo. Por medio de esta relación se podrá identificar a qué nivel pertenece el nodo.
- Estado. Activo o Inactivo.

Por medio de este formulario el usuario podrá:

- Crear un nodo.
- Actualizar un nodo.
- Crear un nodo a partir de otro nodo existente.

Nota: Para crear un nodo se tiene como precondition que el tipo de nodo al cual pertenece este creado y se encuentre en estado activo.

2.5.5 Portlet maestroPlantillasPQPorlet

Con el fin de reutilizar los modelos a emplear al momento de evaluar un producto, bien o servicio, se creó el portlet maestroPlantillasPQPorlet, el cual implementa el caso de uso CU-07.

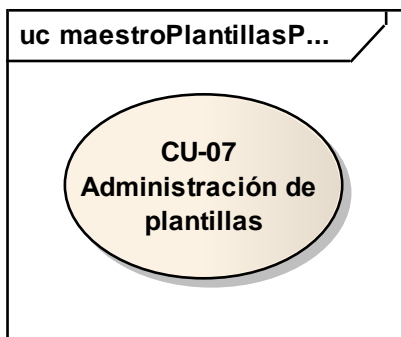


Figura 18. maestroPlantillasPQPortlet.²⁵

²⁵ Elaboración propia.

2.5.6 CU-07 Administración de plantillas

Esta funcionalidad el permite al usuario adicionar nuevos modelos, editar los modelos existentes, y en aquellos casos que se requiera, inactivar los modelos para que no sean usados en los futuros proyectos de evaluación.

Nota: Previamente a la creación de una plantilla se deben haber creado todos los diferentes nodos que la componen.

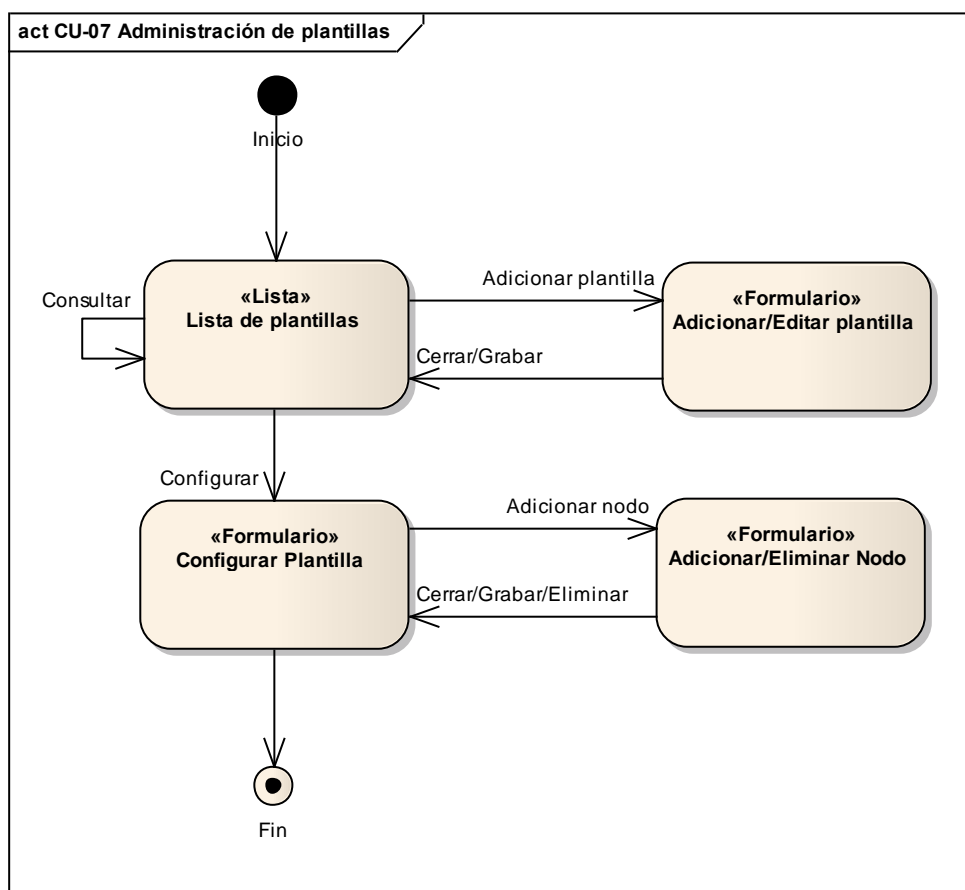


Figura 19. Administración de plantillas.²⁶

²⁶ Elaboración propia.

Plantilla [X]

Identificador
SQuaRE 25010

Nombre
SQuaRE 25010

Descripción
Modelo de calidad, propuesto por la ISO, para evaluar la calidad de un producto software, ya sea de manera interna o externa.

Estado
Activo [v]

Grabar

Figura 21. Adicionar/Editar plantilla.²⁸

Una vez el usuario da click en “Adicionar plantilla” o selecciona uno de los registros y presiona el botón editar, se carga un formulario con los siguientes campos:

- Identificador. Código asignado a la plantilla.
- Nombre.
- Descripción.
- Estado. Activo o Inactivo.

Por medio de este formulario el usuario podrá:

- Crear una plantilla.
- Modificar la información básica de una plantilla.
- Crear una plantilla con la información básica de otra.

²⁸ Elaboración propia.



Figura 22. Configurar plantilla.²⁹

Luego de seleccionar una plantilla en la lista y presionar el botón configurar, el sistema le carga al usuario la pestaña “Configurar plantilla” la cual permite administrar los nodos que pertenecen al modelo. Esta pestaña está compuesta por:

- Tree. Componente que permite recorrer todo el modelo.
- Una lista que contiene los nodos que son hijos del nodo seleccionado en el Tree.

²⁹ Elaboración propia.

Nodo [X]

Plantilla
SQuaRE 25010

Nodo padre
Interoperabilidad

Nodo
Conformidad con la interoperabilidad

Grabar Eliminar

Figura 23. Adicionar/Eliminar nodo.³⁰

Para adicionar o eliminar un nodo hijo a un nodo, el usuario debe seguir los siguientes pasos:

1. Dar click sobre el nodo padre en el componente tree ubicado a la izquierda.
2. Presionar “Adicionar nodo” o dar click sobre uno de los registros de la lista.
3. Presionar grabar o eliminar según la acción que se desee efectuar (Eliminar o crear).

Nota: Si se selecciona un nodo y en el formulario se modifica el campo nodo, el sistema adiciona el nuevo nodo hijo sin eliminar el seleccionado inicialmente.

2.5.7 Portlet maestroProyectosPQPortlet

Este portlet permite soportar las siguientes funcionalidades:

- Crear/Editar proyectos
- Finalizado el proceso de evaluación, el Líder gestor haciendo uso de este portlet podrá lanzar la tarea para calcular la calificación final asignada al producto.

³⁰ Elaboración propia.

- Adicionar el análisis realizado al resultado final.
- Visualizar el modelo empleado para la evaluación.
- Administrar los usuarios que participarán en el proceso de evaluación del proyecto.
- Configurar cada nodo del modelo. Indicar el peso de cada nodo, definir los valores esperados de cada nodo.

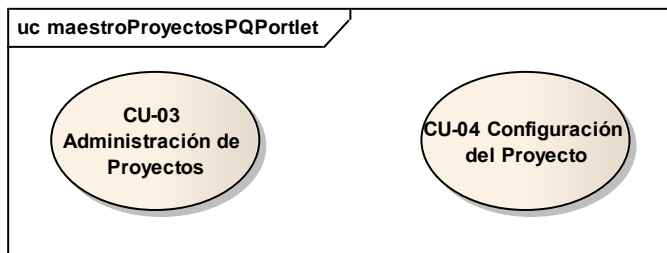


Figura 24. Casos de usos.³¹

³¹ Elaboración propia.

Maestro de proyectos

Maestro proyecto

Adicionar proyecto

Identificador	Nombre	Estado	Calificación
PR01	Modulo de Cartera	Activo	
SQ001-04	Sistema Único de Beneficios (SUB)	Activo	7
SQ001-05	Buscador de referentes bibliográficos	Activo	6
SAP001	Módulo SAP SD	Activo	0
SQ001-06	Empresa Multinacional de Desarrollo De Software - EMDS	Activo	7
SQ001-03	Plan de Acción y Comportamientos	Activo	7
2510	PR 2510	Activo	7
SQ001-01	La Flota	Activo	
SQ001-02	Simulador	Activo	8

< < > >

9 registros, mostrando 9 registro(s), desde 1 hasta 9. Pagina 1 / 1.

Configurar

Editar

Guest - Maestro de proyectos - Maestro de proyectos

Figura 25. Lista de proyectos.³²

³² Elaboración propia.

Proyecto

Identificador

SQ001-04

Nombre

Sistema Único de Ber

Calificación

7

Descripción

El cual es un proyecto desarrollado por CONSULT-SOFT S.A. para la Alcaldía de Medellín y permite el registro y asignación de beneficios subciados por las diferentes secretarías

Conclusión

Estado

Activo

Calificar

Grabar

Figura 26. Adicionar, editar, calificar proyecto.³³

³³ Elaboración propia.

Maestro de proyectos

☐ Maestro proyecto
 ☒ Modelo
 ☐ Usuarios

[Actualizar modelo](#)

- PR 2510
 - Fiabilidad
 - Funcionalidad
 - Interoperabilidad
 - Mantenibilidad
 - Portabilidad
 - Seguridad
 - Usabilidad
 - Eficiencia

Proyecto
PR 2510

Nodo padre

Nodo
Fiabilidad

% Pendiente por asignar
0

Descripción
La capacidad del producto de software para mantener un nivel específico de funcionamiento cuando se está utilizando bajo condiciones especificadas.

Peso %
10

Valor esperado
10

Calificación
7

Conclusión

Grabar

Guest - Maestro de proyectos - Maestro de proyectos

Figura 27. Configurar nodos del modelo. Adicionar conclusión al resultado obtenido por nodo.³⁴

³⁴ Elaboración propia.

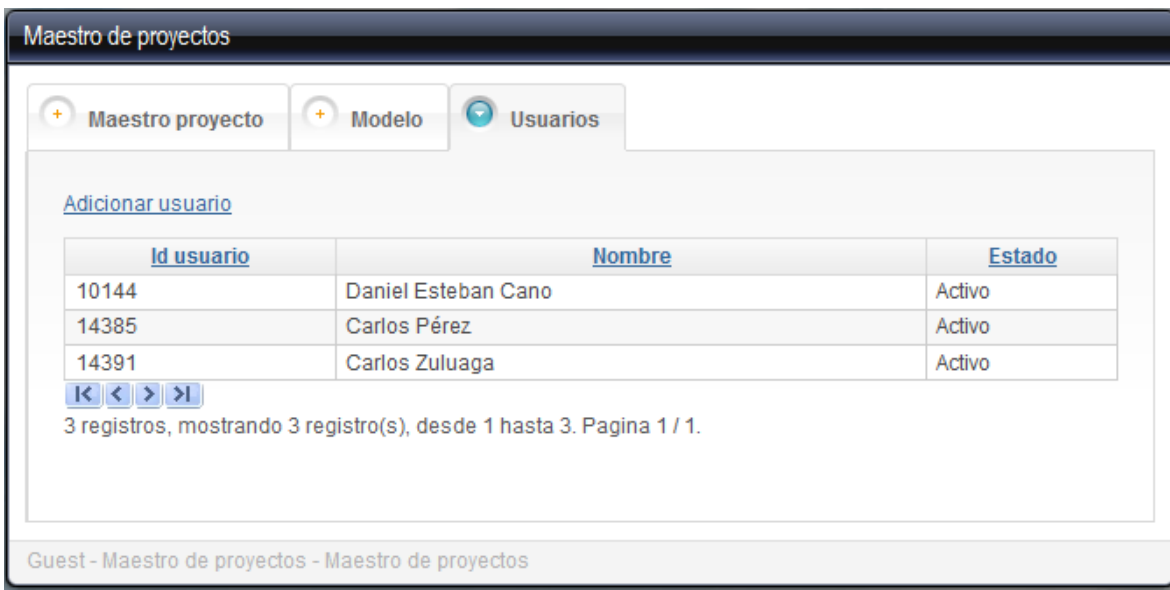


Figura 28. Calificadores asignados al proyecto.

The screenshot shows a form titled 'Usuario' with a red 'X' icon in the top right corner. The form contains the following fields:

- Usuario:** A text input field containing 'Daniel Esteban Cano'.
- Estado:** A dropdown menu with 'Activo' selected.
- Grabar:** A button at the bottom right of the form.

Figura 29. Adicionar/Editar usuario.³⁵

2.5.8 Portlet calificacionPQPortlet

Por medio de este portlet, los usuarios con perfil de evaluador podrán asignarle un valor cuantitativo a cada hoja del modelo. Para el caso del modelo SQuaRE 25010, esta funcionalidad le permitirá al usuario registrar la calificación asignada a cada atributo, teniendo en cuenta que para calcular el valor de la misma se debe seguir la métrica que acompaña a cada atributo.

³⁵ Elaboración propia.

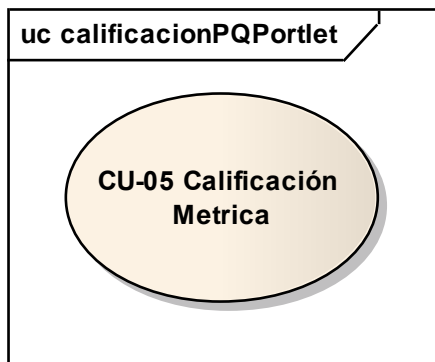


Figura 30. Casos de uso que implementa el portlet calificacionPQPortlet.³⁶

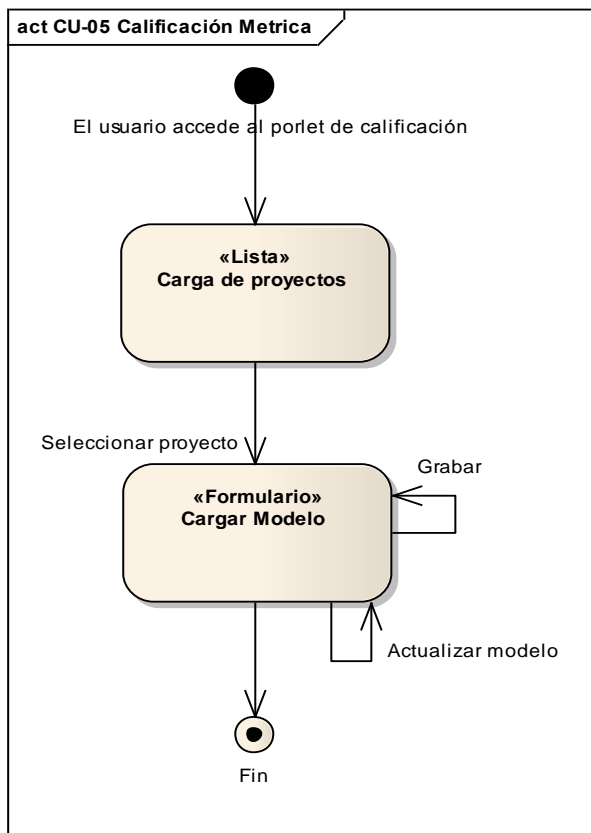


Figura 31. Actividades calificar modelo.³⁷

³⁶ Elaboración propia

³⁷ Elaboración propia.

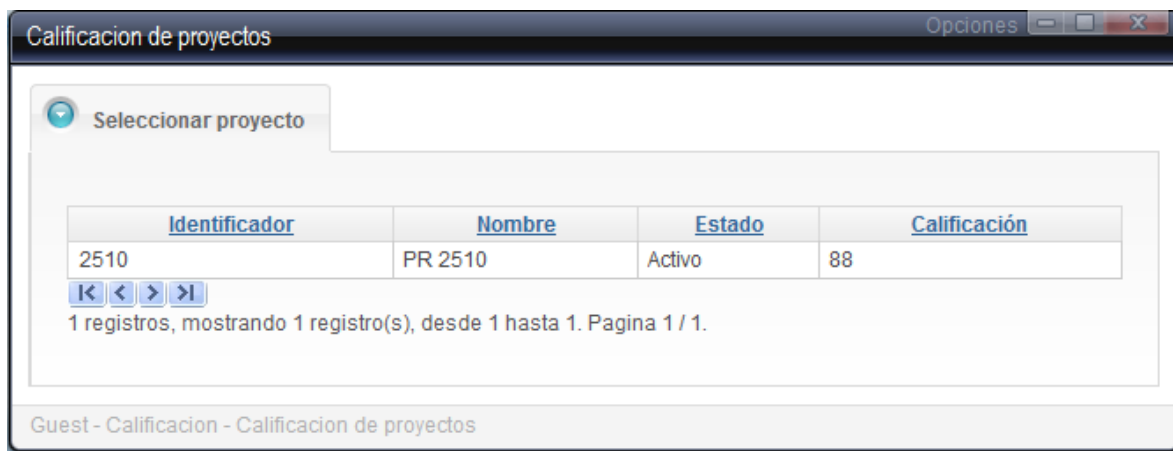


Figura 32. Carga de proyectos.³⁸

En la pestaña “Seleccionar proyecto”, se carga el listado de proyectos a los cuales tiene acceso el usuario.

³⁸ Elaboración propia.

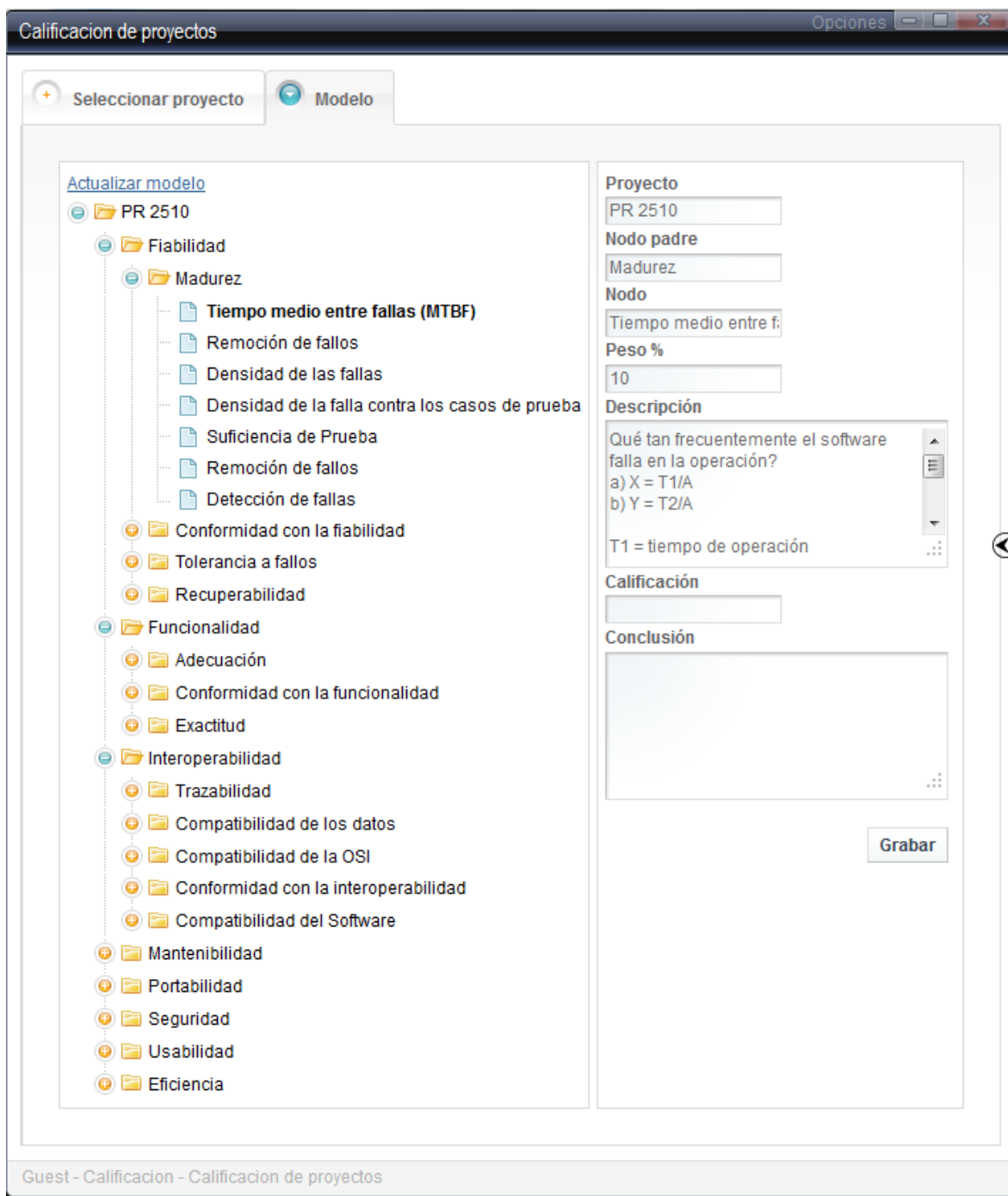


Figura 33. Selección del atributo.³⁹

³⁹ Elaboración propia.

Al dar click sobre alguno de los registros de la lista que aparece en la opción “Seleccionar proyecto“, el sistema carga la pestaña modelo, en la cual el usuario cuenta con las siguientes secciones.

- **Modelo.** En esta sección se encuentra un componente que le permite al usuario navegar por todo el modelo y así seleccionar las hojas que va a calificar.
- **Formulario.** Una vez se selecciona un nodo del modelo, el sistema carga los siguientes campos:
 - **Proyecto.** Contiene el código del proyecto.
 - **Nodo padre.** Nombre del nodo padre.
 - **Nodo.** Nombre del nodo seleccionado.
 - **Peso.** Importancia que se le dio al nodo en el nodo padre.
 - **Descripción.** Descripción del nodo.
 - **Calificación.** En este campo el usuario asigna la calificación a la hoja del árbol-modelo (En el modelo SQuaRE 25010 sería el atributo). La nota asignar se debe encontrar entre 0 y 5, siendo 5 el cumplimiento total de la métrica y 0 el incumplimiento de la misma.
 - **Conclusión.** En este campo el usuario adiciona el análisis realizado a la calificación asignada a la hoja.

2.5.9 Casos de uso vs Requisitos funcionales

A continuación se muestra la relación de casos de uso y requisitos funcionales. Con esto se busca lo siguiente:

- **Trazabilidad.** Al modificar un requisito es posible identificar que casos de uso son afectados y en definitiva determinar el nivel de impacto que conlleva el cambio.
- **Alcance.** Determinar que las funcionalidades a construir suplen las necesidades planteadas.

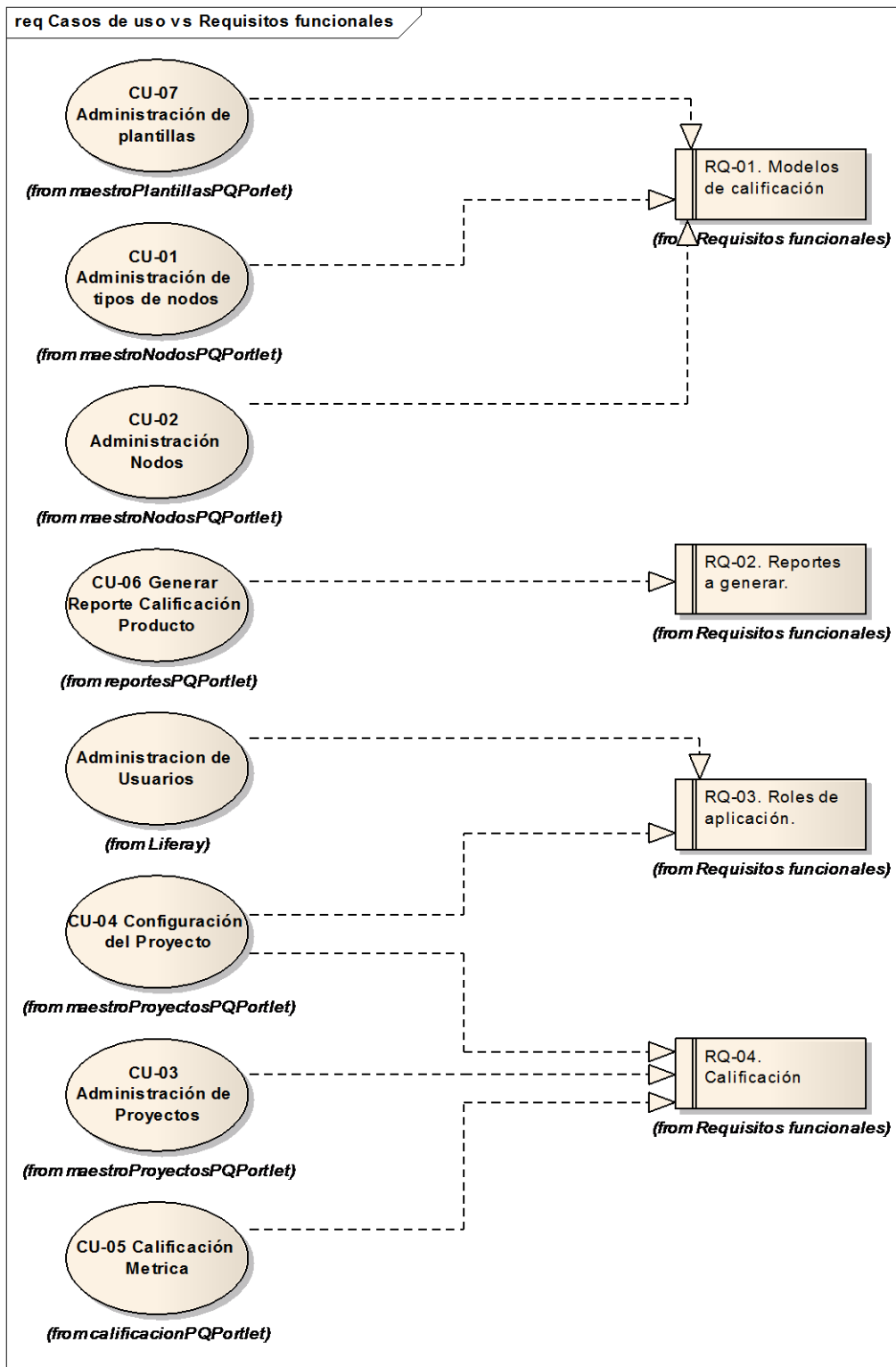


Figura 34. Casos de uso vs requisitos funcionales.⁴⁰

⁴⁰ Elaboración propia.

A partir del anterior diagrama se puede determinar lo siguiente:

- Los casos de uso CU-01, CU-02, CU-07 realizan el requisito funcional RQ-01.
- El caso de uso CU-06 realiza el requisito funcional RQ-02.
- El sistema de seguridad que provee Liferay y el maestro que permite administrar los usuarios por proyecto (CU-04), realizan el requisito funcional RQ-03.
- La configuración del proyecto (CU-04), el caso de uso CU-03 y CU-05 realizan el requisito funcional RQ-04.

2.6 METODOLOGÍA DE TRABAJO

Antes de iniciar la construcción del producto se buscará adquirir conocimiento sobre estándares internacionales relacionados con la evaluación de productos y servicios de software. En esta fase se busca plantear una solución general que posibilite manejar modelos de evaluación basados en dichos estándares. Como resultado de esta etapa en el presente documento se ilustra a grandes rasgos las funcionalidades que va a soportar la aplicación.

En la metodología de construcción de los diferentes portlets se han definido los siguientes pasos a seguir:

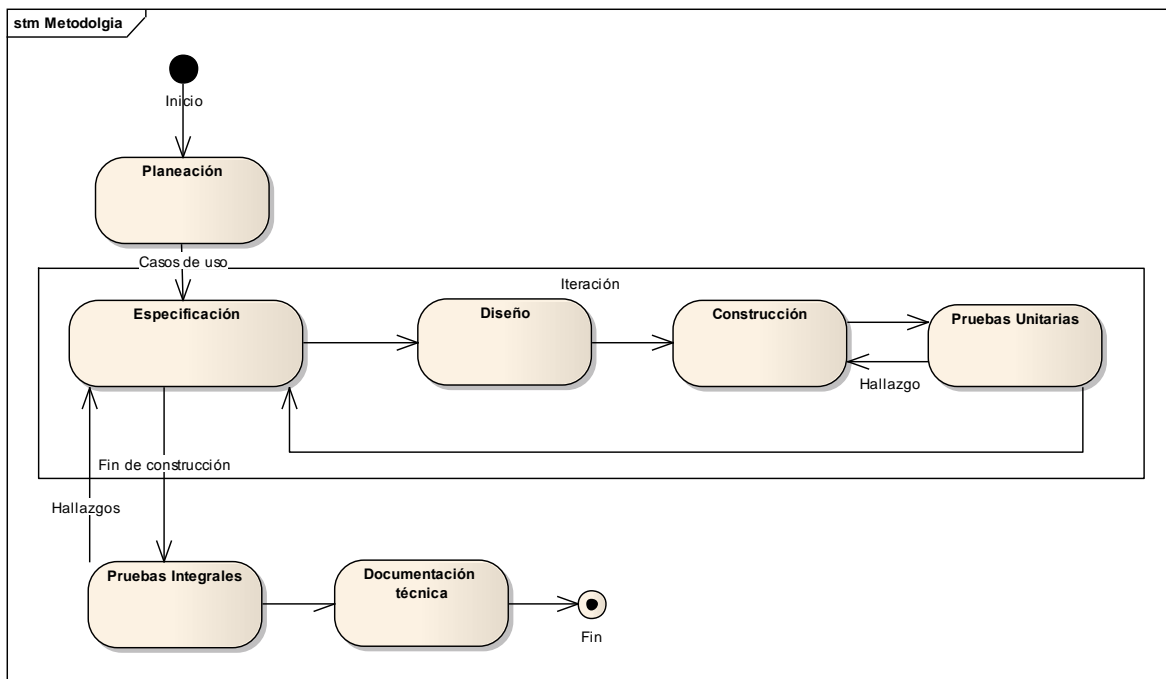


Figura 35. Metodología de trabajo.⁴¹

- Arquitectura. Construcción de un portlet base (maestroNodosPQPortlet), que posibilite la creación de los demás portlets.
- Planeación. En esta etapa se definen los siguientes temas:
 - Orden en que serán construidas las diferentes funcionalidades que provee el sistema.
 - Hitos. Fechas de entrega de los Portlets. Para definir las diferentes fechas de entrega se consideró el tiempo requerido para desarrollar cada opción y la disponibilidad de los recursos del proyecto.
- Iteración. Conjunto de pasos ejecutados al momento de construir cada portlet.

⁴¹ Elaboración propia.

- Especificación. Como resultado de ejecutar esta etapa se obtendrá la descripción detallada de la funcionalidad que provee el portlet.
- Diseño. Se planea a nivel macro cómo se va a efectuar la construcción del portlet.
- Construcción.
- Pruebas Unitarias. Se van a ejecutar las siguientes actividades:
 - Crear casos de prueba con Junit.
 - Hacer una prueba funcional finalizada la construcción de cada portlet.
- Pruebas Integrales. Estas pruebas serán realizadas una vez se ejecuten todas las iteraciones.
 - Probar todas las funcionalidades.
 - Hacer un piloto con un grupo de estudiantes.
- Documentación técnica. Haciendo uso de la herramienta Enterprise Architect, se hará reingeniería para generar la documentación técnica del producto.

Como resultado final se entregarán los siguientes artefactos:

- Documentación del proyecto. Este documento consolida lo siguiente:
 - Especificación de todas las funcionalidades que provee el producto.
 - Documentación técnica del producto (Diagrama de clases, modelo entidad relación y diagrama de componentes).
- Archivos war. Portlets construidos.

2.7 DESPLIGUE DE LA APLICACIÓN

Para realizar el despliegue de SERVIQUALITY se deben seguir los siguientes pasos.

- Descargar Liferay 5.2.3. La versión “Community edition” ya se encuentra configurada con el servidor de aplicaciones que se desee emplear. Para las pruebas el servidor de aplicaciones que vamos a emplear es Tomcat.
- Se debe instalar el motor de base de datos relacional que se desea emplear (MySQL, Oracle, PostGreSQL,...).
- Se debe configurar la conexión a base de datos en el servidor de aplicaciones.
- Los portlets podrán ser instalados por medio de la consola administrativa del portal o copiando los archivos .war en la carpeta “deploy”.

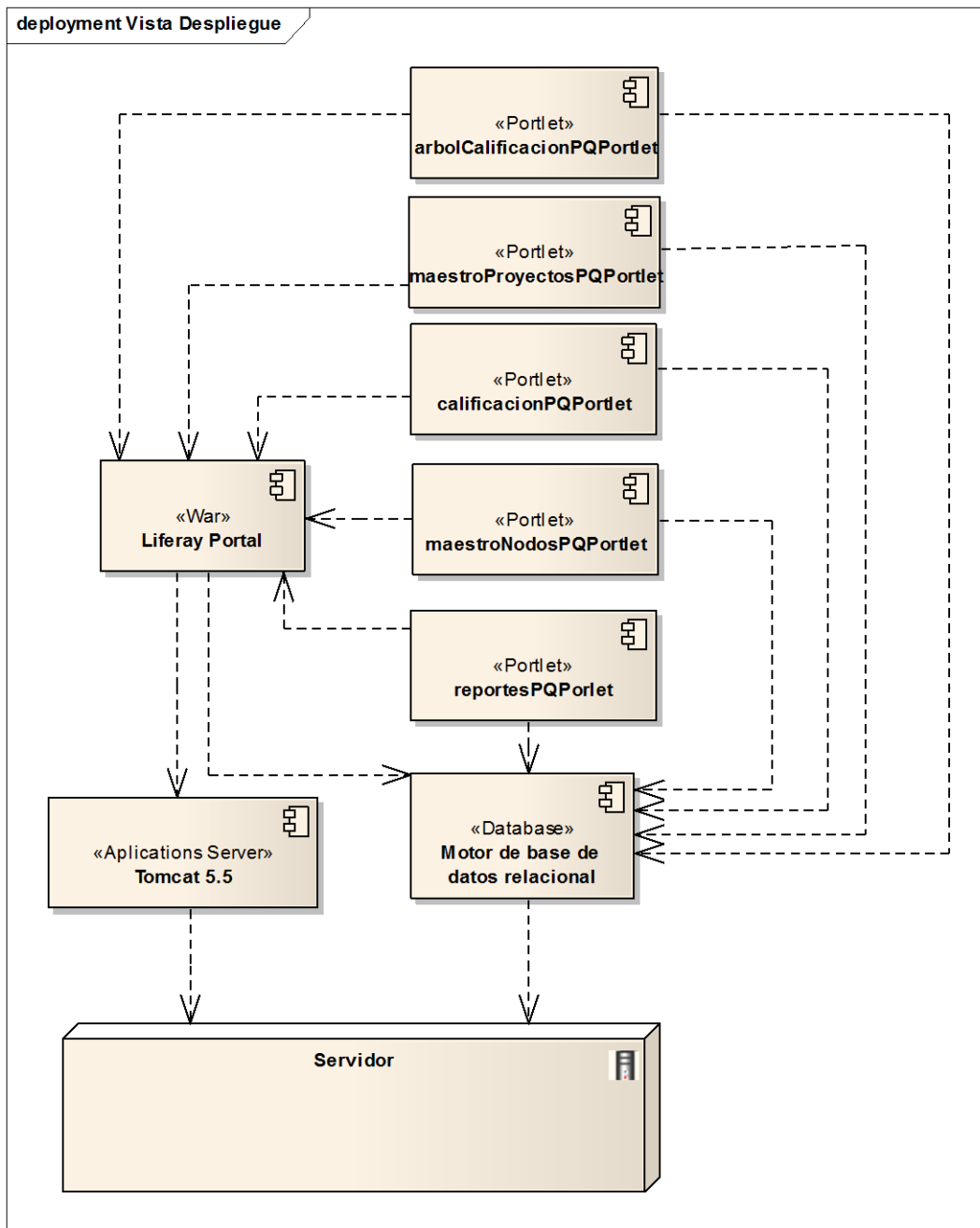


Figura 36. Diagrama de despliegue.⁴²

Como se puede observar en el anterior diagrama tenemos lo siguiente:

- Los portlets dependen de Liferay y un motor de base de datos relacional.

⁴² Elaboración propia

- Para funcionar Liferay requiere un motor de base de datos relacional y un servidor de aplicaciones.
- Tanto el motor de base de datos como el servidor de aplicaciones requieren de un Sistema operativo para funcionar (Windows, Linux o Unix.).

2.8 AMBIENTE DE DESARROLLO – PRUEBAS

A continuación se describen las diferentes herramientas empleadas para la construcción y puesta a disposición de SERVIQUALITY.

Para la construcción de Serviquality se emplearon las siguientes herramientas:

- Eclipse Helios. IDE de desarrollo liviano que permite crear diferentes espacios de trabajo y ser integrado a Liferay para la creación de portlets.
- Tomcat 5.5. Servidor de aplicaciones. Servidor de aplicaciones liviano que puede ser instalado en servidores con sistemas operativos Linux, Windows y Unix.
- Oracle 11g. Base de datos relacional. Uno de los principales motores de base de datos relacionales empleados a nivel corporativo.
- Java 6. Leguaje de programación.
- Windows 7. Sistema operativo. Sistema operativo con el cual cuenta el equipo en el cual se va a realizar la construcción de la aplicación
- Liferay 5.2. Portal Web. En el momento que se inicio el proyecto esta era la última versión “community” que se tenía a disposición.

Con el fin de poner a disposición un ambiente de pruebas al cual pueda acceder toda la comunidad interesada, se adquirió un dominio y se realizó la configuración de la infraestructura correspondiente (Configuración en Router, DNS) para posibilitar el uso de los portlets.

Las herramientas empleadas para el ambiente de prueba, son:

- Tomcat 5.5. Servidor de aplicaciones liviano que ya viene integrado con Liferay 5.2.3.
- Oracle 11g. Principal base de datos relacional empleada a nivel corporativo.
- Java 6. Leguaje de programación.
- Liferay 5.2. Portal Web. En el momento que se inicio el proyecto esta era la última versión “community” que se tenía a disposición.
- Windows Server 2008. Uno de los principales sistema operativos empleados a nivel corporativo. Se elije este sistema operativo dado que es fácil de configurar y provee varias herramientas como Servidor DNS, Active directory, las cuales serán de gran utilidad para poner a disposición SERVIQUALITY.
- Active directory. Principal repositorio de usuarios empleado a nivel corporativo.

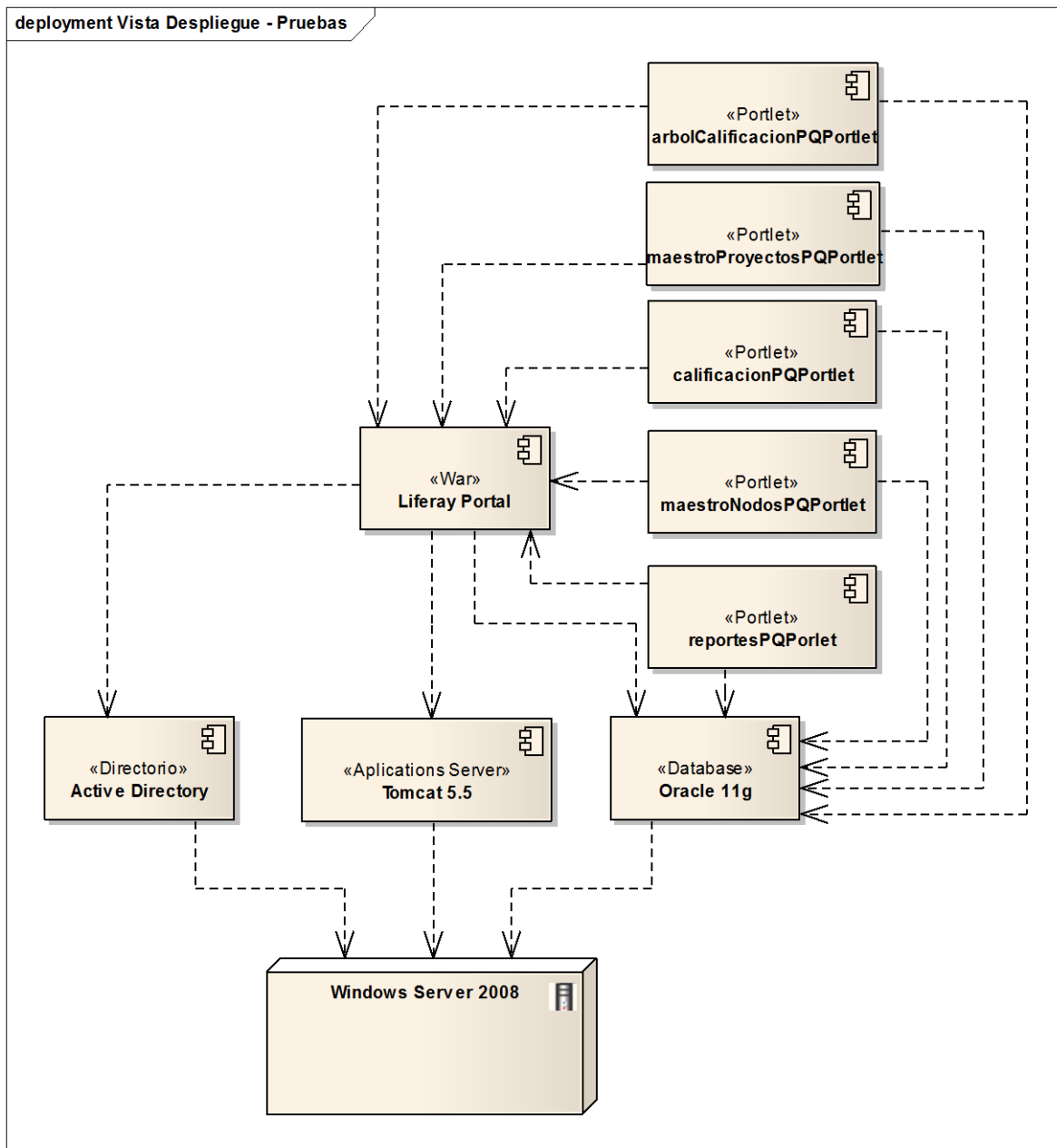


Figura 37. Diagrama de despliegue.⁴³

⁴³ Elaboración propia.

2.9 ARQUITECTURA DE COMPONENTES

El diseño de la solución se encuentra basado en el uso de varios patrones, entre los que se encuentra MVC y DAO. La solución maneja las siguientes capas:

- **Presentación.** En esta capa se encuentran todos los componentes de Front End, los cuales son los encargados de interactuar con el usuario. Estos componentes hacen uso de la capa de negocio para procesar los datos enviados por el usuario.
- **Negocio.** En esta capa se encuentran los componentes de Back end, los cuales contienen toda la lógica de negocio. Estos componentes hacen uso de la capa DAO, para interactuar con la Base de datos y así dar respuesta a las solicitudes enviadas desde la capa de presentación.
- **DAO.** Los componentes de esta capa son los directamente responsables de la interacción que se efectúe sobre la Base de datos. Se hace uso de Hibernate para brindar una mayor portabilidad a la solución final.
- **Modelo.** Es una capa transversal a las capas de presentación, negocio y DAO. Esta capa contiene la declaración de los objetos que van a ser intercambiados por estas 3 capas.
- **Utilidades.** Esta capa contiene los componentes que ofrecen un conjunto de utilidades para facilitar la interacción entre las capas Presentación, Negocio y DAO.

- Base de datos. Esta Capa contiene todos los objetos de Base de datos que participan en la solución; en este caso son secuencias y tablas.

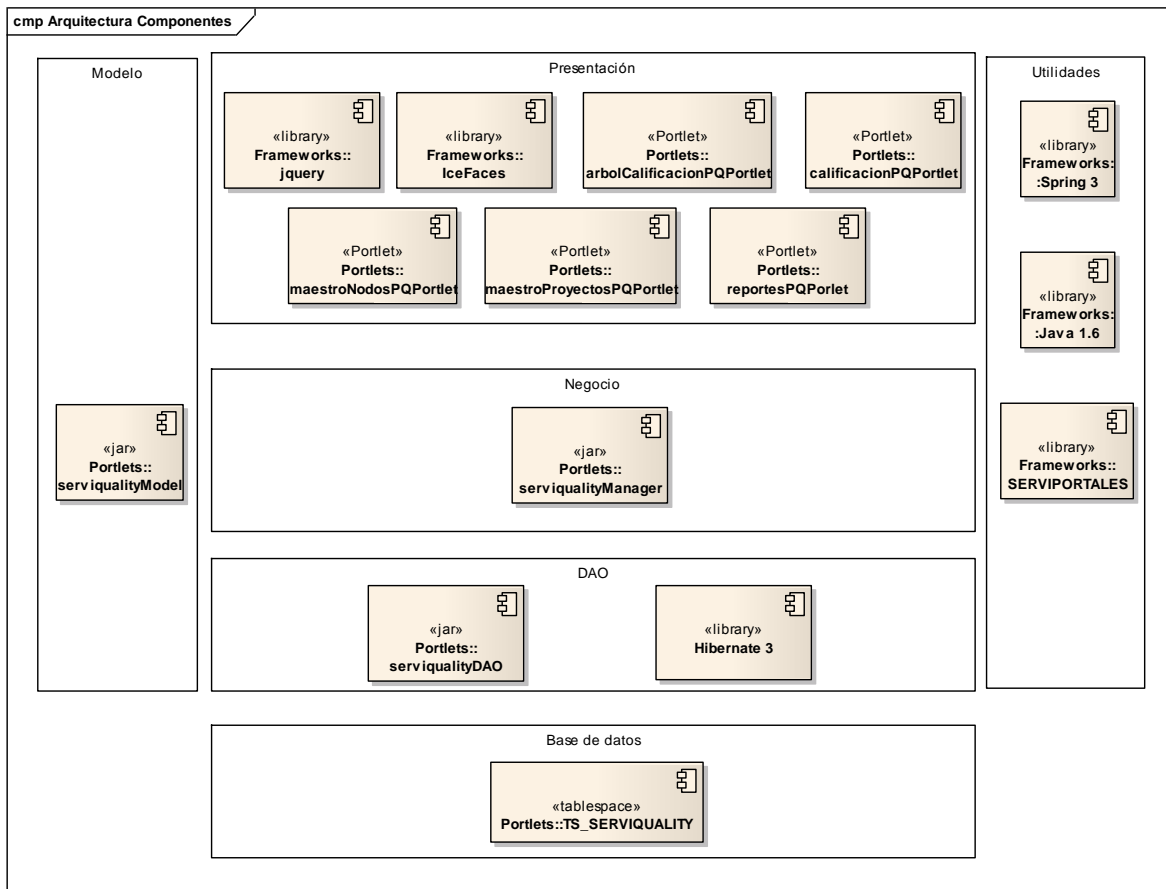


Figura 38. Arquitectura de componentes.⁴⁴

⁴⁴ Elaboración propia.

2.10 MODELO DE CLASES – ENTIDADES

Con las clases Entidad se busca mapear las Tablas de base de datos. Las Clases que pertenecen al paquete PK como son NodoPK, PlantillaNodoPK, ProyectoNodoPK, ProyectoUserPK, son empleadas para mapear las claves primarias compuestas (la clave primaria de la tabla es conformada por varias columnas).

Tabla en Base de datos	Clase entidad
TSQ_CAL_PROYECTO_NODO	CalProyectoNodo
TSQ_NODO	Nodo y NodoPK
TSQ_PLANTILLA	Plantilla y Árbol
TSQ_PLANTILLA_NODO	PlantillaNodo y PlantillaNodoPK
TSQ_PROYECTO	Proyecto y Árbol
TSQ_PROYECTO_NODO	ProyectoNodo y ProyectoNodoPK
TSQ_PROYECTO_USER	ProyectoUser y ProyectoUserPK
TSQ_TIPO_NODO	TipoNodo

Tabla 1. Tablas vs Clases entidad.

Las clases Plantilla y Proyecto heredan de la clase Árbol, permitiendo manejar una información común y otra particular al proyecto o la plantilla.

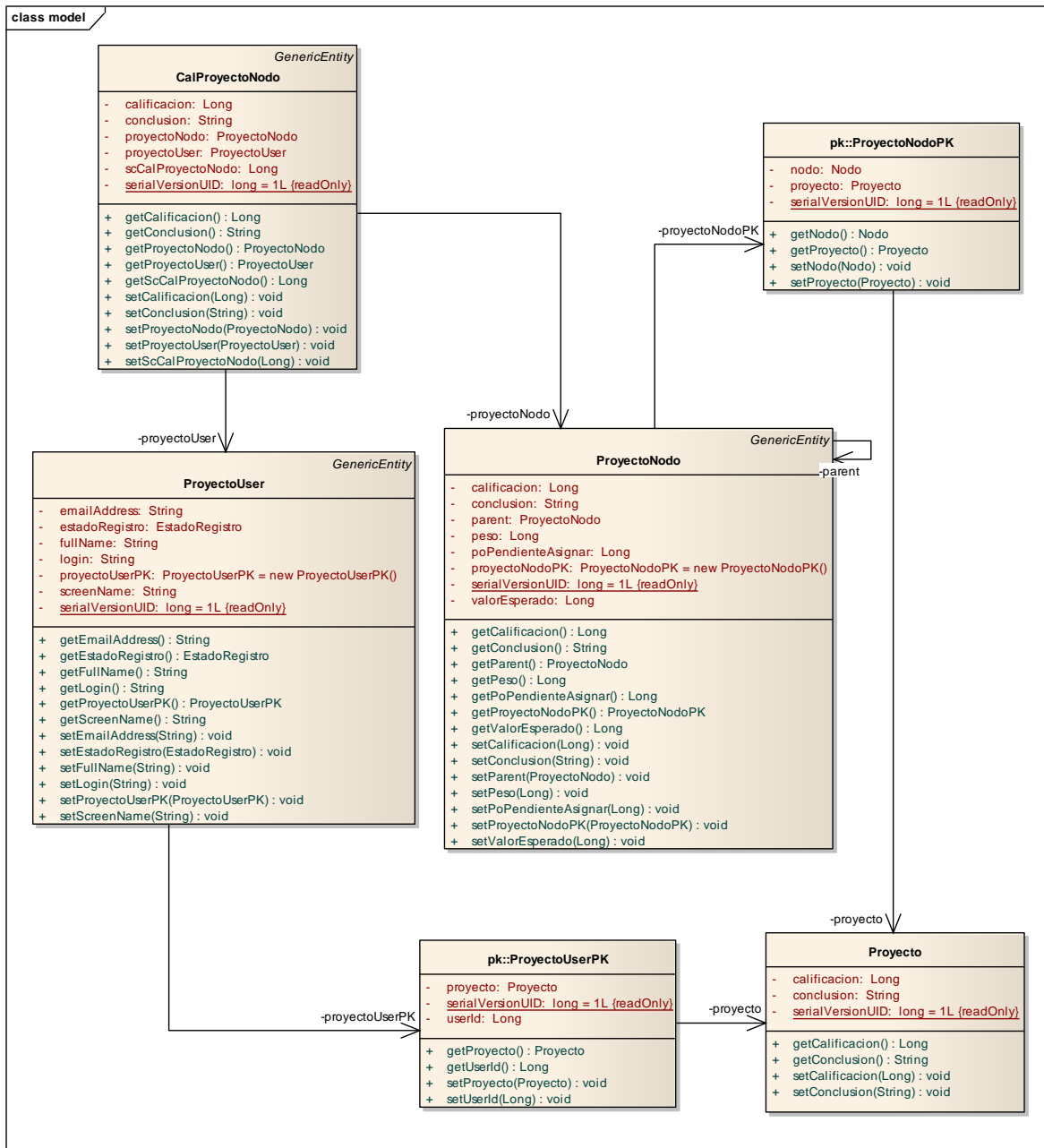


Tabla 2. Modelo de clases - Entidades. Parte 1.⁴⁵

⁴⁵ Elaboración propia.

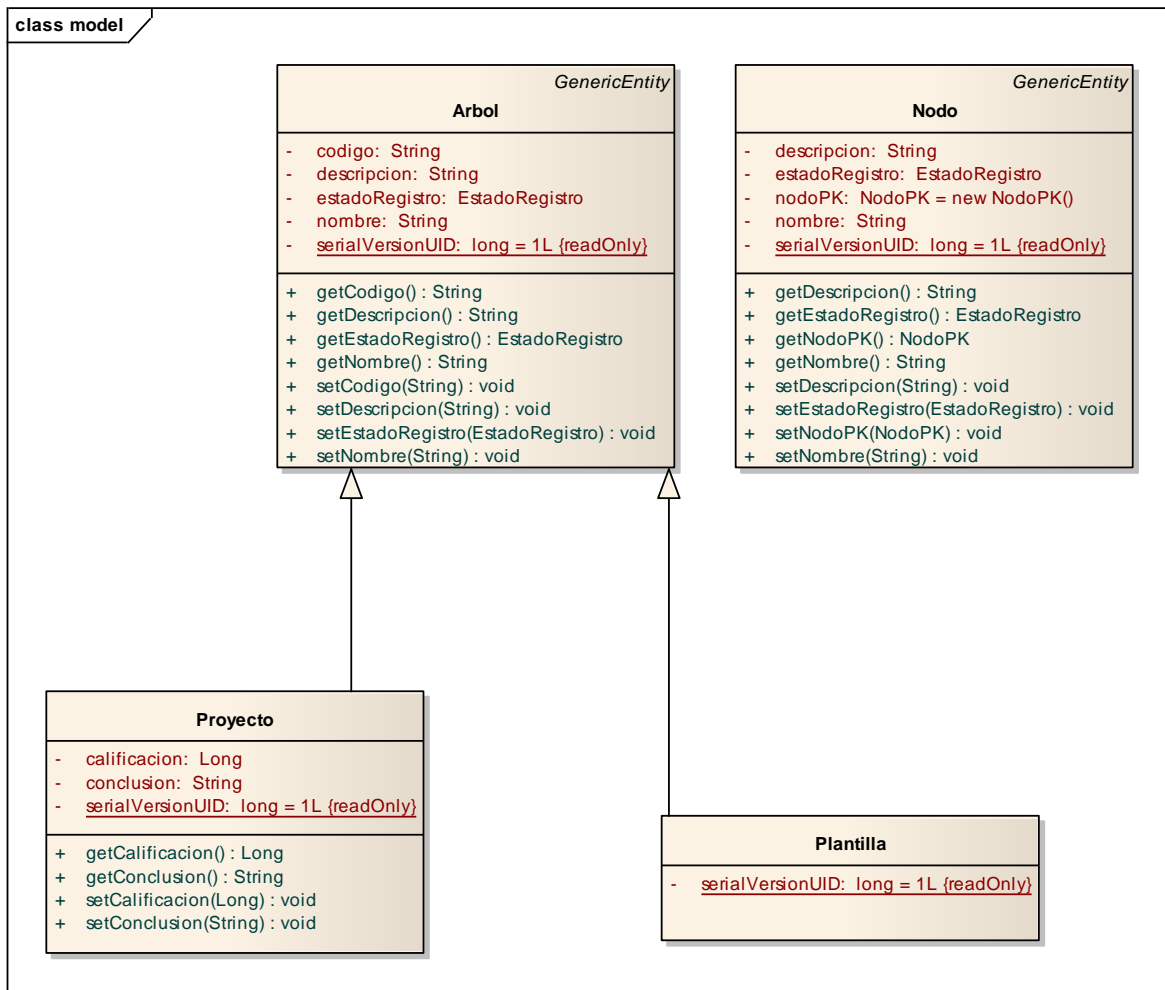


Tabla 3. Modelo de clases - Entidades. Parte 2.⁴⁶

⁴⁶ Elaboración propia.

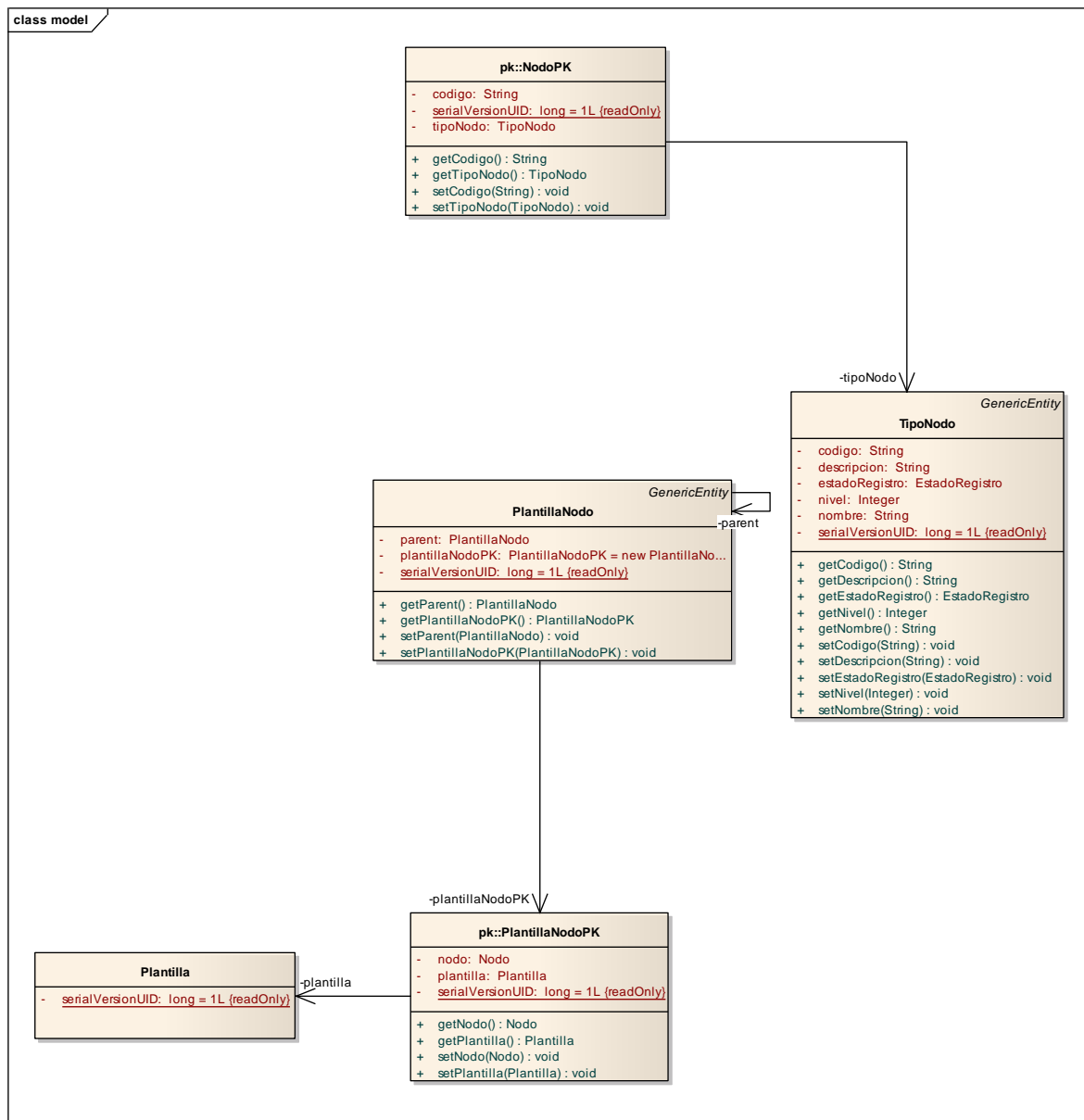


Tabla 4. Modelo de clases - Entidades. Parte 3.⁴⁷

⁴⁷ Elaboración propia.

2.11 MODELO DE DATOS

Para explicar el modelo de datos de SERVIQUALITY, este ha sido dividido según su uso en 3 grupos.

Modelo de datos. Plantilla.

Este grupo de tablas contiene los datos administrados por medio de los portlets maestroNodosPQPortlet y maestroPlantillasPQPortlet.

TSQ_TIPO_NODO. Contiene los niveles a emplear en los diferentes modelos.

PK	Nombre	Tipo	Not Null	Unique	Len	Pre c	Scal e	Init	Observaciones
True	CODIGO	VARC HAR2	False	True	20				Código asignado al nivel.
False	DESCRIPCION	VARC HAR2	False	False	400				Campo de observación.
False	ESTADO_REGISTRO	VARC HAR2	False	False	10				Estado del registro. Activo "A", Inactivo "I".
False	NIVEL	NUMB ER	False	False		10	0		Nivel.
False	NOMBRE	VARC HAR2	False	False	120				Nombre otorgado al nivel.

Tabla 5. TSQ_TIPO_NODO – Columnas.

Nombre	Tipo	Columnas	Initial Code	Observaciones
SYS_C0015572	Public	CODIGO		Clave primaria.

Tabla 6. TSQ_TIPO_NODO – Constraints.

Columnas	Asociación	Observaciones
(CDTIPO_NODO CODIGO)	= 0..* TSQ_NODO.FKCC1FD879456038CF 1 TSQ_TIPO_NODO.SYS_C0015572	

Tabla 7. TSQ_TIPO_NODO - Relationships.

TSQ_NODO. Contiene el grupo de nodos empleados para estructurar los modelos.

PK	Nombre	Tipo	Not Null	Unique	Length	Pre c	Scale	Init	Observaciones
True	CODIGO	VARC HAR2	False	True	20				Código asignado al nodo.
False	DESCRIPCION	VARC HAR2	False	False	700				Campo de observación.
False	ESTADO_REGISTRO	VARC HAR2	False	False	10				Estado del registro. Activo A o Inactivo I
False	NOMBRE	VARC HAR2	False	False	120				Nombre asignado al nodo
True	CDTIPO_NODO	VARC HAR2	False	True	20				Código asignado al nivel al cual pertenece el nodo.

Tabla 8. TSQ_NODO - Columnas.

Nombre	Tipo	Columnas	Initial Code	Observaciones
SYS_C0015556	Public	CODIGO CDTIPO_NODO		
FKCC1FD879456038CF	Public	CDTIPO_NODO		

Tabla 9. TSQ_NODO – Constraints

Columnas	Asociación	Observaciones
----------	------------	---------------

Columnas	Asociación	Observaciones
(CDTIPO_NODO = CODIGO)	0..* TSQ_NODO.FKCC1FD879456038CF 1 TSQ_TIPO_NODO.SYS_C0015572	
(CDNODO = CODIGO CDTIPO_NODO = CDTIPO_NODO)	0..* TSQ_PLANTILLA_NODO.FK839D40551 CBAF1E8 1 TSQ_NODO.SYS_C0015556	
(CDNODO = CODIGO CDTIPO_NODO = CDTIPO_NODO)	0..* TSQ_PROYECTO_NODO.FKC7D109391 CBAF1E8 1 TSQ_NODO.SYS_C0015556	

Tabla 10. TSQ_NODO - Relationships.

TSQ_PLANTILLA_NODO. Almacena la estructura de los diferentes modelos.

PK	Nombre	Tipo	Not Null	Unique	Length	Pre c	Scale	Init	Observaciones
True	CDARBOL	VARC HAR2	False	True	100				Código asignado al modelo
True	CDNODO	VARC HAR2	False	True	20				Código asignado al nodo.
True	CDTIPO_NODO	VARC HAR2	False	True	20				Código asignado al nivel al cual pertenece el nodo.
False	CDNODO_PARENT	VARC HAR2	False	False	20				Código del nodo padre.
False	CDTIPO_NODO_PARENT	VARC HAR2	False	False	20				Nivel asignado al nodo Padre.
False	CDARBOL_PARENT	VARC HAR2	False	False	100				Código del modelo al cual pertenece el nodo padre.

Tabla 11. TSQ_PLANTILLA_NODO – Columns.

Nombre	Tipo	Columnas	Initial Code	Observaciones
SYS_C0015560	Public	CDARBOL CDNODO CDTIPO_NODO		
FK839D40551CBAF1E8	Public	CDNODO CDTIPO_NODO		
FK839D4055AA3DA197	Public	CDARBOL		
FK839D4055E70096E1	Public	CDNODO_PARENT CDTIPO_NODO_PARENT CDARBOL_PARENT		

Tabla 12. TSQ_PLANTILLA_NODO - Constraints.

Columnas	Asociación	Observaciones
(CDNODO = CODIGO CDTIPO_NODO = CDTIPO_NODO)	0..* TSQ_PLANTILLA_NODO.FK839D4055 1CBAF1E8 1 TSQ_NODO.SYS_C0015556	
(CDARBOL = CODIGO)	0..* TSQ_PLANTILLA_NODO.FK839D4055 AA3DA197 1 TSQ_PLANTILLA.SYS_C0015559	
(CDNODO_PARENT = CDNODO CDTIPO_NODO_PAR ENT = CDTIPO_NODO CDARBOL_PARENT = CDARBOL)(CDNODO	0..* TSQ_PLANTILLA_NODO.FK839D4055 E70096E1 1 TSQ_PLANTILLA_NODO.SYS_C00155 60	

Columnas	Asociación	Observaciones
_PARENT = CDARBOL CDTIPO_NODO_PARENT = CDNODO CDARBOL_PARENT = CDTIPO_NODO)		

Tabla 13. TSQ_PLANTILLA_NODO - Relationships.

TSQ_PLANTILLA. Contiene la información básica (Código, Nombre, Descripción) de cada modelo.

PK	Nombre	Tipo	Not Null	Unique	Length	Pre c	Scale	Init	Observaciones
True	CODIGO	VARC HAR2	False	True	100				Código asignado al modelo
False	DESCRIPCION	VARC HAR2	False	False	700				Campo de observación.
False	ESTADO_REGISTRO	VARC HAR2	False	False	10				Estado del registro, activo A o inactivo I.
False	NOMBRE	VARC HAR2	False	False	200				Nombre asignado al modelo

Tabla 14. TSQ_PLANTILLA – Columnas

Nombre	Tipo	Columnas	Initial Code	Observaciones
SYS_C0015559	Public	CODIGO		

Tabla 15. TSQ_PLANTILLA - Constraints.

Columnas	Asociación	Observaciones
(CDARBOL = CODIGO)	0..* TSQ_PLANTILLA_NODO.FK839D4055A A3DA197 1 TSQ_PLANTILLA.SYS_C0015559	

Tabla 16. TSQ_PLANTILLA - Relationships.

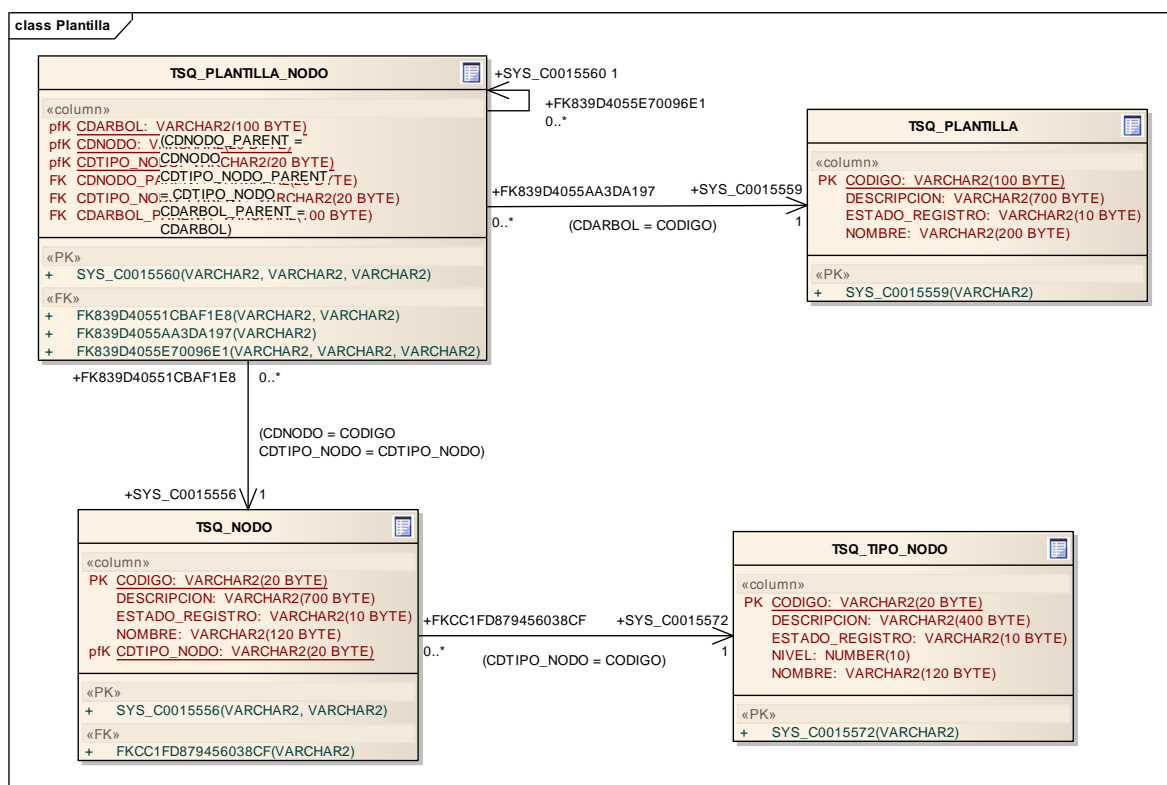


Figura 40. Modelo de datos. Plantilla.⁴⁹

Modelo de datos. Proyecto.

El conjunto de tablas que se ilustran a continuación contiene los datos diligenciados por medio del portlet `mestroProyectosPQPortlet`.

⁴⁹ Elaboración propia.

TSQ_PROYECTO_NODO. Una vez se elige una plantilla para el proyecto el sistema recrea la estructura del modelo haciendo uso de esta tabla.

PK	Nombre	Tipo	Not Null	Unique	Length	PREC	SCALE	Init	Observaciones
False	CONCLUSION	VARCHAR2	False	False	700				Campo de observación.
False	PESO	NUMBER	False	False		19	0		Nivel de importancia otorgado al nodo (0-100%)
True	CDARBOL	VARCHAR2	False	True	100				Código asignado al modelo.
True	CDNODO	VARCHAR2	False	True	20				Código asignado al nodo.
True	CDTIPO_NODO	VARCHAR2	False	True	20				Nivel al cual pertenece el nodo.
False	CDNODO_PARENT	VARCHAR2	False	False	100				Código nodo padre
False	CDTIPO_NODO_PARENT	VARCHAR2	False	False	20				Nivel al cual pertenece el nodo padre.
False	CDARBOL_PARENT	VARCHAR2	False	False	20				Código del modelo al cual pertenece el nodo padre.
False	VALOR_ESPERADO	NUMBER	False	False		19	0		Nota final que se espera obtener

									para el nodo.
Fals e	CALIFICACION	NUMBER	False	False		19	0		Nota final obtenida.

Tabla 17. TSQ_PROYECTO_NODO - Columnas.

Nombre	Tipo	Columnas	Initial Code	Observaciones
SYS_C0015564	Public	CDARBOL CDNODO CDTIPO_NODO		
FKC7D109391CBAF1E8	Public	CDNODO CDTIPO_NODO		
FKC7D10939F2FE71AB	Public	CDARBOL		
FKC7D1093943F33CF5	Public	CDNODO_PARENT CDTIPO_NODO_PARENT CDARBOL_PARENT		

Tabla 18. TSQ_PROYECTO_NODO – Constraints

Columnas	Asociación	Observaciones
(CDARBOL = CODIGO)	0..* TSQ_PROYECTO_NODO.FKC7D10939F2FE71AB 1 TSQ_PROYECTO.SYS_C0015563	
(CDNODO = CODIGO CDTIPO_NODO = CDTIPO_NODO)	0..* TSQ_PROYECTO_NODO.FKC7D109391CBAF1E8 1 TSQ_NODO.SYS_C0015556	
(CDNODO = CDNODO CDTIPO_NODO = CDTIPO_NODO)	0..* TSQ_CAL_PROYECTO_NODO.FKFB384E89DE5866B	

Columnas	Asociación	Observaciones
CDARBOL_NODO = CDARBOL)(CDNODO = CDARBOL CDTIPO_NODO = CDNODO CDARBOL_NODO = CDTIPO_NODO)	1 TSQ_PROYECTO_NODO.SYS_C0015 564	
(CDNODO_PARENT = CDNODO CDTIPO_NODO_PAR ENT = CDTIPO_NODO CDARBOL_PARENT = CDARBOL)(CDNODO _PARENT = CDARBOL CDTIPO_NODO_PAR ENT = CDNODO CDARBOL_PARENT = CDTIPO_NODO)	0..* TSQ_PROYECTO_NODO.FKC7D1093 943F33CF5 1 TSQ_PROYECTO_NODO.SYS_C0015 564	

Tabla 19. TSQ_PROYECTO_NODO - Relationships.

TSQ_PROYECTO. Contiene la información básica (Código, Nombre, Descripción) de cada proyecto de calificación.

PK	Nombre	Tipo	Not Null	Unique	Len	Pre c	Scale	Init	Observaciones
True	CODIGO	VARC	False	True	100				Código asignado

		HAR2							al proyecto.
False	DESCRIPCION	VARC HAR2	False	False	700				Campo de observación.
False	ESTADO_REGISTRO	VARC HAR2	False	False	10				Estado del registro. Activo A o Inactivo I.
False	NOMBRE	VARC HAR2	False	False	200				Nombre del proyecto
False	CALIFICACION	NUMBER	False	False		19	0		Calificación promedio otorgada al nodo.
False	CONCLUSION	VARC HAR2	False	False	700				Campo de observación

Tabla 20. TSQ_PROYECTO - Columnas.

Nombre	Tipo	Columnas	Initial Code	Observaciones
SYS_C0015563	Public	CODIGO		

Tabla 21. TSQ_PROYECTO - Constraints.

Columnas	Asociación	Observaciones
(CDARBOL CODIGO) =	0..* TSQ_PROYECTO_NODO.FKC7D10939F2FE71AB 1 TSQ_PROYECTO.SYS_C0015563	
(CDARBOL CODIGO) =	0..* TSQ_PROYECTO_USER.FKC7D446F8F2FE71AB 1 TSQ_PROYECTO.SYS_C0015563	

Tabla 22. TSQ_PROYECTO - Relationships.

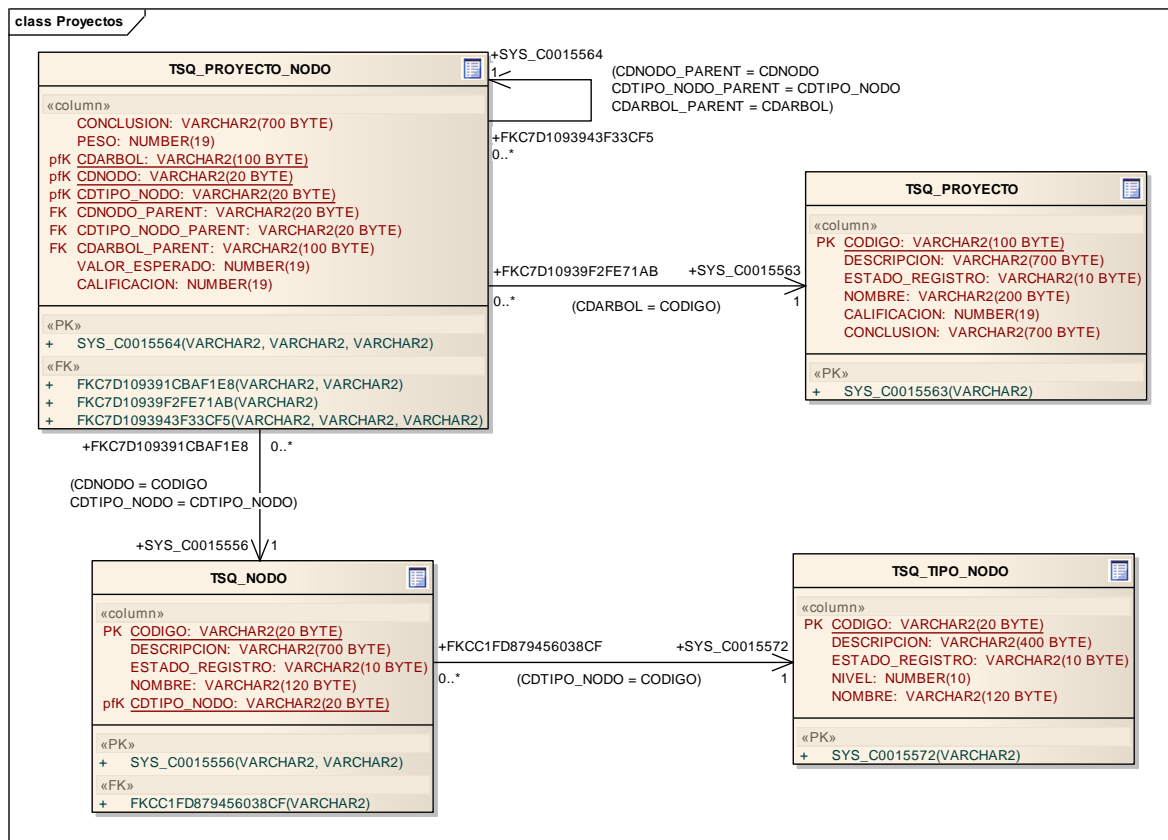


Figura 41. Modelo de datos. Proyectos.⁵⁰

Modelo de datos. Calificación.

Por medio de este conjunto de tablas se puede lograr lo siguiente:

- El maestroProyectosPQPorlet puede almacenar los usuarios que tienen acceso a cada proyecto.
- Es posible registrar la calificación otorgada a cada hoja del modelo haciendo uso del portlet maestroCalificaciónPQPortlet.

En la tabla TSQ_PROYECTO_USER, se encuentran los códigos internos de los usuarios matriculados en el portal, Liferay, que tienen acceso a la información del proyecto.

⁵⁰ Elaboración propia.

PK	Nombre	Tipo	Not Null	Unique	Length	Pre c	Scale	Init	Observaciones
True	USER_ID	NUMBER	False	True		19	0		Código interno del usuario que hace parte del proyecto.
False	ESTADO_REGISTRO	VARCHAR2	False	False	10				Estado del registro. Activo "A", Inactivo "I".
True	CDARBOL	VARCHAR2	False	True	100				Código del proyecto

Tabla 23. TSQ_PROYECTO_USER - Columnas.

Nombre	Tipo	Columnas	Initial Code	Observaciones
SYS_C0015567	Public	USER_ID CDARBOL		
FKC7D446F8F2FE71AB	Public	CDARBOL		

Tabla 24. TSQ_PROYECTO_USER - Constraints.

Columnas	Asociación	Observaciones
(CDARBOL CODIGO)	= 0..* TSQ_PROYECTO_USER.FKC7D446F8F2FE71AB 1 TSQ_PROYECTO.SYS_C0015563	
(CDARBOL_USER CDARBOL USER_ID USER_ID)(CDARBOL_USER = USER_ID)	= 0..* TSQ_CAL_PROYECTO_NODO.FKFB384E833710915 1 TSQ_PROYECTO_USER.SYS_C0015	

Columnas	Asociación	Observaciones
USER_ID (CDARBOL)	= 567	

Tabla 25. TSQ_PROYECTO_USER - Relationships.

En la tabla TSQ_CAL_PROYECTO_NODO, Se almacena la nota otorgada por cada Calificador a las diferentes hojas (En el modelo Square 25010 hace referencia a los atributos) que componen el modelo empleado para la calificación del producto o servicio.

PK	Nombre	Tipo	Not Null	Unique	Length	Precision	Scale	Init	Observaciones
True	SCCALPROYECTONODO	NUMBER	False	True		19	0		Código interno que se le otorga a la calificación almacenada
False	CALIFICACION	NUMBER	False	False		19	0		Nota de 0 a 10 asignada al nodo.
False	CONCLUSION	VARCHAR2	False	False	700				Campo de observación.
False	CDNODO	VARCHAR2	False	False	100	0	0		Código asignado al nodo.
False	CDTIPO_NODO	VARCHAR2	False	False	20	0	0		Código asignado al nivel al que pertenece el nodo.
False	CDARBOL_MODEL	VARCHAR2	False	False	20	0	0		Código asignado al modelo

Falso	CDARBOL_US ER	NUMB ER	False	False	0	19	0		Código asignado al modelo.
Falso	USER_ID	VARC HAR2	False	False	100	0	0		Código interno del usuario que realiza la calificación

Tabla 26. TSQ_CAL_PROYECTO_NODO - Columnas.

Nombre	Tipo	Columnas	Initial Code	Observaciones
SYS_C0015551	Public	SCCALPROYECTO ONODO		
SYS_C0015552	Public	CDNODO CDTIPO_NODO CDARBOL_NODO CDARBOL_USER USER_ID		
FKFB384E89DE5866 B	Public	CDNODO CDTIPO_NODO CDARBOL_NODO		
FKFB384E833710915	Public	CDARBOL_USER USER_ID		

Tabla 27. TSQ_CAL_PROYECTO_NODO - Constraints.

Columnas	Asociación	Observaciones
(CDARBOL_USER = CDARBOL USER_ID = USER_ID)(CDARBOL _USER = USER_ID USER_ID =	0..* TSQ_CAL_PROYECTO_NODO.FKFB3 84E833710915 1 TSQ_PROYECTO_USER.SYS_C0015 567	

Columnas	Asociación	Observaciones
CDARBOL)		
(CDNODO = CDNODO CDTIPO_NODO = CDTIPO_NODO CDARBOL_NODO = CDARBOL)(CDNODO = CDARBOL CDTIPO_NODO = CDNODO CDARBOL_NODO = CDTIPO_NODO)	0..* TSQ_CAL_PROYECTO_NODO.FKFB3 84E89DE5866B 1 TSQ_PROYECTO_NODO.SYS_C0015 564	

Tabla 28. TSQ_CAL_PROYECTO_NODO - Relationships.

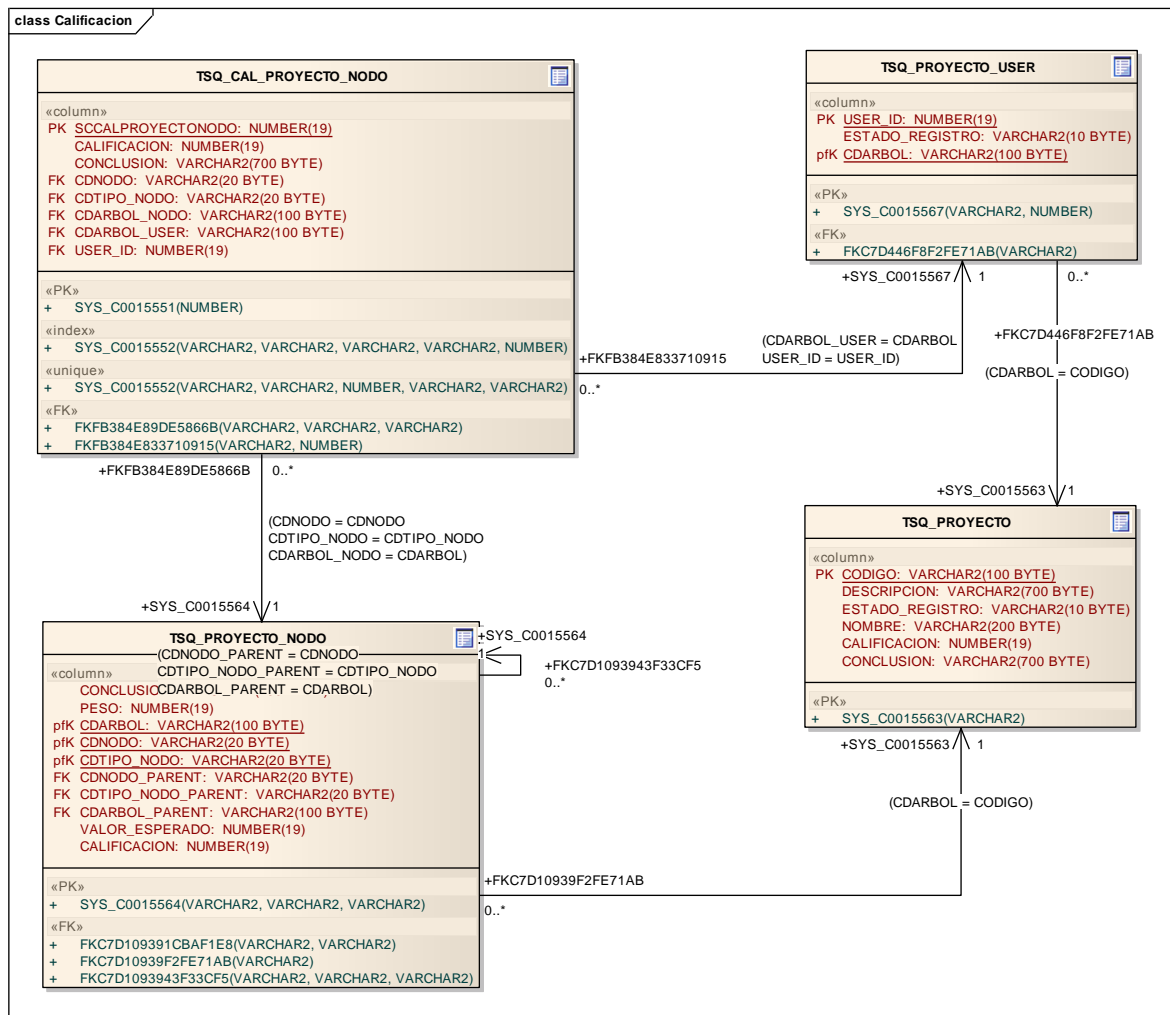


Figura 42. Modelo de datos. Calificación.⁵¹

⁵¹ Elaboración propia.

CONCLUSIONES Y OPORTUNIDADES DE MEJORA

SERVIQUALITY permite calificar adicional a servicios y productos software otros tipos de producto (Autos, aviones, trenes, vías nacionales,...); solamente se requiere contar con un modelo (Basado en una estructura de Árbol) que posibilite la calificación del producto deseado.

Al emplear el modelo propuesto por la norma SQuaRE 25010 para calificar un producto software, se busca que el resultado del mismo sea cuantitativo. Se ha encontrado que este proceso sigue teniendo cierto grado de subjetividad dado que el Líder gestor determina el nivel de importancia para cada Característica, Subcaracterísticas y Atributo. Es por este motivo que se propone que los modelos a implementar, entre los cuales se encuentra la norma SQuaRE 25010, estén acompañados de un instructivo o aplicación software que permita orientar al Líder gestor en la definición del nivel de importancia a otorgar a cada componente del modelo a emplear.

Haciendo uso del sistema, se han identificado los siguientes aspectos por mejorar:

- Dado que los usuarios con perfil de Evaluador deberán diligenciar la calificación final otorgada a la métrica, se propone que el sistema permita indicar la fórmula empleada para calcular el valor de la métrica; de esta manera los usuarios con perfil de Evaluador se deberán enfocar en obtener el valor de cada variable. Con esto se busca agilizar la ejecución de las calificaciones por parte del usuario.
- Con miras a convertir SERVIQUALITY en una herramienta base para la mejora continua, se sugiere que pueda contar con reportes estadísticos que permitan visualizar el comportamiento de los productos en el tiempo (nivel de calidad) y así apoyar la toma de decisiones.

- Poner a disposición una opción que le permita al Líder gestor visualizar las notas y conclusiones que ha adicionado cada evaluador del proyecto.
- Adicionar una funcionalidad que permita exportar las conclusiones realizadas a cada nodo del proyecto.
- Permitir el ingreso de valores con puntos decimales en los campos “calificación” y “valor esperado”.
- Dado que un producto puede ser calificado varias veces hasta que cumpla con la calidad esperada para pasar a un ambiente productivo, es necesario permitir hacer cierres de evaluación que se realiza al mismo producto para posteriormente poder hacer comparaciones contra resultados anteriores e identificar el avance que se ha tenido en la calidad del producto.

BIBLIOGRAFÍA

algoritmia.net. algoritmia.net. [En línea] [Citado el: 26 de Febrero de 2012.] <http://www.algoritmia.net/articles.php?id=17>.

Carvalho, Juan Pablo y Franch, Xavier. 2003. *Using Quality Models Software Package Selection*. Cataluña : Universidad Politécnica de Cataluña, 2003.

Franco Córdoba, Ana María y Jaramillo Ospina, Luz Adriana. 2003. *Guía técnica sobre evaluación de software para la administración pública*. Medellín : Universidad EAFIT, 2003.

International Standards for Business, Government and Society. 2001. *Information technology - Software Product Quality- Part 1: Quality model*. s.l. : ISO and IEC, 2001. ISO/IEC 9126-1.

International Standards for Business, Government and Society. 2007. *Software engineering - Software product Quality Requirements and Evaluation (SQuare) - quality model*. s.l. : ISO and IEC, 2007. ISO/IEC 25010.

Juristo, Natalia, Moreno, Ana M y Vegas, Sira. 2004. *Técnicas de evaluación de software*. 2004.

Mendoza, Luis E, Pérez, María A y Grimán, Anna C. 2005. *Prototipo de modelo sistémico de calidad (MOSCA) del software*. s.l. : Universidad Simón Bolívar, 2005.

Piedrahita Mesa, Sebastián. 2007. *Construcción de una herramienta para evaluar la calidad de un producto software*. Medellín : Universidad EAFIT, 2007.

Paz, Angel Cervera. Monografias. [En línea] [Citado el: 14 de 04 de 2012.]
<http://www.monografias.com/trabajos5/call/call.shtml>.

Resolución ministerial. 2004. *Guía técnica sobre evaluación de software para la administración pública.* s.l. : Presidencia del Consejo de Ministros, 2004. n° 139-2004-pcm.

Fillottrani, Pablo R. Departamento de ciencias é ingeniería de la computación.
Universidad nacional del sur. Buenos Aires. Argentina. [En línea] [Citado el: 14 de 04 de 2012.] <http://www.cs.uns.edu.ar/~prf/teaching/SQ07/clase6.pdf>

ANEXO A – TÉRMINOS Y DEFINICIONES

Campo autocomplete.

Los campos autocomplete muestran sugerencias en la medida que el usuario escribe.

TI

Es necesario establecer que la tecnología de la información (TI) se entiende como "aquellas herramientas y métodos empleados para recabar, retener, manipular o distribuir información. La tecnología de la información se encuentra generalmente asociada con las computadoras y las tecnologías afines aplicadas a la toma de decisiones (Bologna y Walsh, 1997: 1).

Portlet

“Los **portlets** son componentes modulares de las interfaces de usuario gestionadas y visualizadas en un portal web. Los *portlets* producen fragmentos de código de marcado que se agregan en una página de un portal. Típicamente, siguiendo la metáfora de escritorio, una página de un portal se visualiza como una colección de ventanas de *portlet* que no se solapan, donde cada una de estas muestra un *portlet*. Por lo tanto un *portlet* (o colección de *portlets*) se asemeja a una aplicación web que está hospedada en un portal. Como por ejemplo, un portlet de aplicación puede ser para el correo, el parte meteorológico, un foro, noticias, etc”⁵².

⁵² **Wikipedia.** Wikipedia. [En línea] [Citado el: 26 de Febrero de 2012.] <http://es.wikipedia.org/wiki/Portlet>.

Liferay

“Es un portal de gestión de contenidos de código abierto escrito en Java. Se creó en 2000 en principio como solución para las organizaciones sin ánimo de lucro”⁵³.

“Un portal es un sitio web cuya característica fundamental es la de servir de *Puerta de entrada* (única) para ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios relacionados a un mismo tema. Incluye: enlaces, buscadores, foros, documentos, aplicaciones, compra electrónica, etc. Principalmente un portal en Internet está dirigido a resolver necesidades de información específica de un tema en particular”⁵⁴.

Árbol – Nodo – Hoja

Un árbol es una estructura de datos, que puede definirse de forma recursiva como, una estructura vacía o un elemento o clave de información (nodo) más un número finito de estructuras tipo árbol, disjuntos, llamados subárboles.

Nomenclatura sobre árboles:

⁵³ **Wikipedia.** Wikipedia. [En línea] [Citado el: 26 de Febrero de 2012.] <http://es.wikipedia.org/wiki/Liferay>.

⁵⁴ **Wikipedia.** Wikipedia. [En línea] [Citado el: 26 de Febrero de 2012.] http://es.wikipedia.org/wiki/Portal_%28Internet%29.

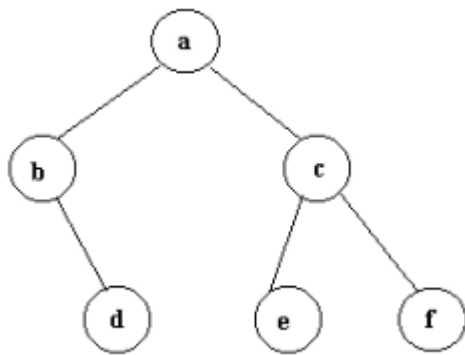


Figura 43. Árbol.⁵⁵

- Raíz: es aquel elemento que no tiene antecesor; ejemplo: *a*.
- Rama: arista entre dos nodos.
- Antecesor: un nodo *X* es antecesor de un nodo *Y* si por alguna de las ramas de *X* se puede llegar a *Y*. Ejemplo: *c* es antecesor de *e*.
- Sucesor: un nodo *X* es sucesor de un nodo *Y* si por alguna de las ramas de *Y* se puede llegar a *X*. Ejemplo: *f* es sucesor de *c*.
- Grado de un nodo: el número de descendientes directos que tiene. Ejemplo: *c* tiene grado 2, *d* tiene grado 0, *a* tiene grado 2.
- Hoja: nodo que no tiene descendientes: grado 0. Ejemplo: *d*
- Nodo interno: aquel que tiene al menos un descendiente.
- Nivel: número de ramas que hay que recorrer para llegar de la raíz a un nodo. Ejemplo: el nivel del nodo *a* es 1 (es un convenio), el nivel del nodo *e* es 3.
- Altura: el nivel más alto del árbol. En el ejemplo de la figura 1 la altura es 3.
- Anchura: es el mayor valor del número de nodos que hay en un nivel. En la figura, la anchura es 3.

⁵⁵ **algoritmia.net.** algoritmia.net. [En línea] [Citado el: 26 de Febrero de 2012.] <http://www.algoritmia.net/articles.php?id=17>.

Para el caso del Modelo SQuaRE - ISO/IEC CD 25010 implementado en SERVIQUALITY vamos a contar con lo siguiente:

- Raíz: Es el nombre asignado a la plantilla o proyecto.
- Nodos o nodos internos: Conjunto de características y Subcaracterísticas.
- Hojas: Conjunto de atributos.

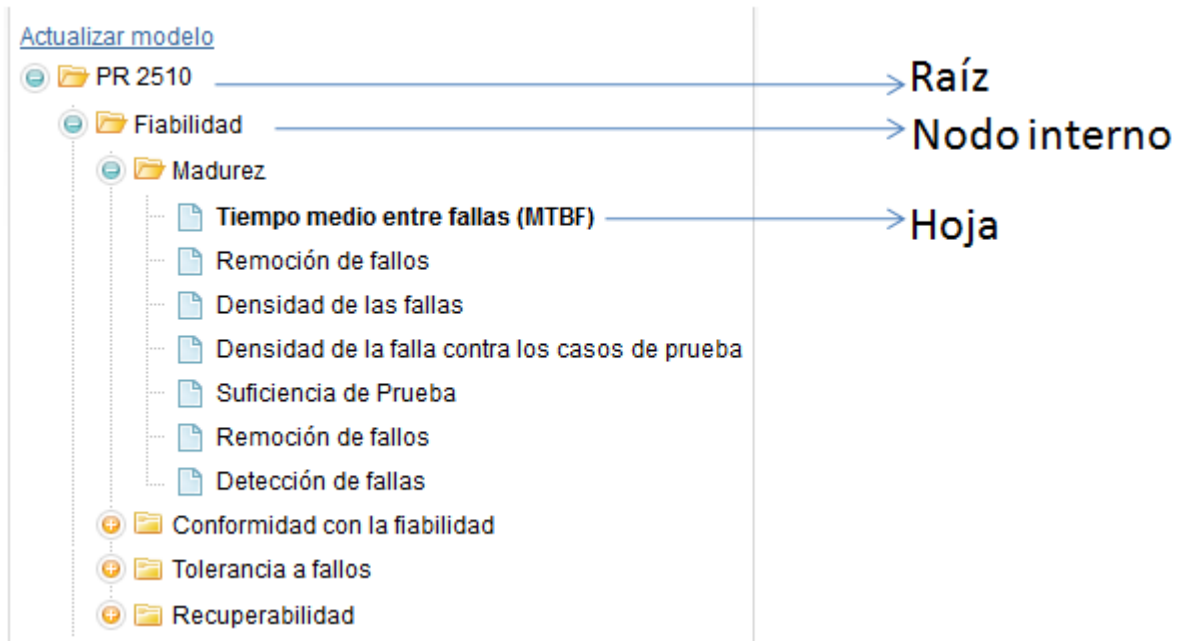


Figura 44. Árbol.⁵⁶

24/7

Es una abreviatura que significa "24 horas al día, 7 días a la semana", refiriéndose usualmente a los negocios o servicios que están disponibles durante todo el tiempo sin interrupción.

Caso de uso – CU

⁵⁶ Elaboración propia.

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. En el contexto de ingeniería del software, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema.

IDE

“Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI)”⁵⁷.

DAO

“**Data Access Object** (DAO, Objeto de Acceso a Datos) es un componente de software que suministra una interfaz común entre la aplicación y uno o más dispositivos de almacenamiento de datos, tales como una Base de datos o un archivo. El término se aplica frecuentemente al Patrón de diseño Object”⁵⁸.

Front-end y back-end

“En diseño de software el front-end es la parte del software que interactúa con el o los usuarios y el back-end es la parte que procesa la entrada desde el front-end.

⁵⁷ **Wikipedia.** Wikipedia. [En línea] [Citado el: 26 de Febrero de 2012.] http://es.wikipedia.org/wiki/Entorno_de_desarrollo_integrado.

⁵⁸ **Wikipedia.** [En línea] [Citado el: 26 de Febrero de 2012.] http://es.wikipedia.org/wiki/Data_Access_Object.

La separación del sistema en "front ends" y "back ends" es un tipo de abstracción que ayuda a mantener las diferentes partes del sistema separadas. La idea general es que el front-end sea el responsable de recolectar los datos de entrada del usuario, que pueden ser de muchas y variadas formas, y procesarlas de una manera conforme a la especificación que el back-end pueda usar. La conexión del front-end y el back-end es un tipo de interfaz”⁵⁹.

Requisitos funcionales

“Un requisito funcional define el comportamiento interno del software: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que muestran cómo los casos de uso serán llevados a la práctica. Son complementados por los requisitos no funcionales, que se enfocan en cambio en el diseño o la implementación.”

“Como se define en la ingeniería de requisitos, los requisitos funcionales establecen los comportamientos del sistema.”⁶⁰

Requisito no funcional

“Un requisito no funcional o atributo de calidad es, en la ingeniería de sistemas y la ingeniería de software, un requisito que especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que éstos corresponden a los requisitos funcionales. Por tanto, se

⁵⁹ **Wikipedia.** Wikipedia. [En línea] [Citado el: 26 de Febrero de 2012.] http://es.wikipedia.org/wiki/Front-end_y_back-end .

⁶⁰ **Wikipedia.** Wikipedia. [En línea] [Citado el: 15 de Abril de 2012.] http://es.wikipedia.org/wiki/Requisito_funcional.

refieren a todos los requisitos que ni describen información a guardar, ni funciones a realizar.”⁶¹

Mantenibilidad

El IEEE define mantenibilidad como: “La facilidad con la que un sistema o componente software puede ser modificado para corregir fallos, mejorar su funcionamiento u otros atributos o adaptarse a cambios en el entorno”.⁶²

E-Learning

“El elearning es un modelo de formación a distancia que utiliza Internet como herramienta de aprendizaje. Este modelo permite al alumno realizar el curso desde cualquier parte del mundo y a cualquier hora.”⁶³

E-Commerce

“Comercio electrónico, también conocido como e-commerce (electronic commerce en inglés), consiste en la compra y venta de productos o de servicios a través de medios electrónicos, tales como Internet y otras redes informáticas.”⁶⁴

⁶¹ **Wikipedia.** Wikipedia. [En línea] [Citado el: 15 de Abril de 2012.] http://es.wikipedia.org/wiki/Requisito_no_funcional.

⁶² **Connexions.** Connexions. [En línea] [Citado el: 22 de Abril de 2012.] <http://cnx.org/content/m17457/latest/>.

⁶³ **Instituto de máquina herramienta.** IMH. [En línea] [Citado el: 22 de Abril de 2012.] <http://www.imh.es/elearning-es/que-es-elearning>

⁶⁴ **Wikipedia.** Wikipedia. [En línea] [Citado el: 22 de Abril de 2012.] http://es.wikipedia.org/wiki/Comercio_electr%C3%B3nico.

ANEXO B – PRUEBA PILOTO

Gracias al apoyo del profesor Rafael Rincón, el producto fue puesto a disposición de un grupo de estudiantes que cursaban la materia Métricas de Software, con el fin de validar la utilidad que representa la herramienta, hacer pruebas funcionales y determinar oportunidades de mejora.

Las herramientas de comunicación empleadas para dar soporte a los estudiantes fueron:

- Chat que provee Liferay.
- Portlet de comentarios que provee Liferay.
- Correo electrónico.

Para efectuar esta prueba piloto se realizaron los siguientes pasos:

Actividad	Responsable
Charla del modelo SQuaRE - ISO/IEC CD 25010.	Rafael Rincón
Charla de funcionamiento de la aplicación.	Daniel Cano
Poner a disposición un ambiente de pruebas	Daniel Cano
Brindar usuarios y claves de acceso al portal	Daniel Cano
Soporte técnico	Daniel Cano
Seguimiento	Rafael Rincón

Tabla 29. Prueba piloto.

Los resultados obtenidos de la prueba piloto, fueron:

- Homologar el producto para que funcione de forma adecuada en diferentes navegadores como son (Internet Explorer 7 o superiores, Mozilla Firefox 4 o superiores, Safari).

- Es necesario aumentar la velocidad de subida en el servidor. Se trabajó con una velocidad 600kbps.
- Aumentar la amigabilidad del sistema. Adicionar videos que posibilite al usuario identificar las acciones que debe seguir para calificar un producto bien o servicio.
- Validar en el servidor los campos diligenciados por el usuario.
- Permitir exportar los resultados obtenidos, dado que luego de haber realizado el proceso de calificación los estudiantes debían hacer entrega de un informe adjuntando los datos ya registrados en las diferentes funcionalidades, lo cual implicaba doble trabajo para los estudiantes.
- Algunos estudiantes no pudieron concluir la calificación debido a que la infraestructura bajo la cual se montó el ambiente de pruebas no contaba con un servicio de 24/7.

Nota: A pesar que varios de los estudiantes decidieron optar por hacer uso de otras herramientas para realizar esta labor, me encuentro satisfecho dado que varios grupos realizaron el proceso utilizando SERVIQUALITY, obteniendo buenos resultados en su trabajo.



<div>  Maestro proyecto </div>			
Adicionar proyecto			
Identificador	Nombre	Estado	Calificación
SQ001-04	Sistema Único de Beneficios (SUB)	Activo	7
SQ001-05	Buscador de referentes bibliográficos	Activo	6
		Activo	
SQ001-06	Empresa Multinacional de Desarrollo De Software - EMDS	Activo	7
SQ001-03	Plan de Acción y Comportamientos	Activo	7
SQ001-02	Simulador	Activo	8
<div>  </div>			
9 registros, mostrando 9 registro(s), desde 1 hasta 9. Pagina 1 / 1.			

Figura 45. Proyectos finalizados satisfactoriamente (5 de 12).⁶⁵

⁶⁵ Elaboración propia.

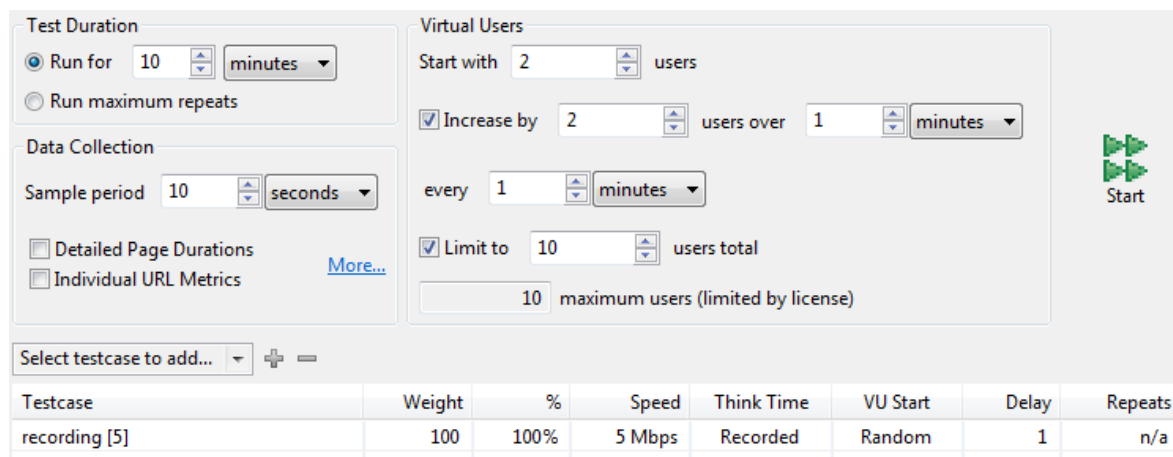
ANEXO C – PRUEBAS NO FUNCIONALES

Finalizada las pruebas pilotos se realizaron varios ajustes al sistema y posterior a esto se ejerció una serie de pruebas de estrés con el fin de validar el cumplimiento de los siguientes requisitos no funcionales:

- RQNF-02 Tiempo de respuesta.
- RQNF-07 Concurrencia.

Para la ejecución de esta prueba se puso a disposición el ambiente empleado para la realización de la prueba piloto.

Haciendo uso de la aplicación “Web performance load tester”, se ejecuto el siguiente escenario:



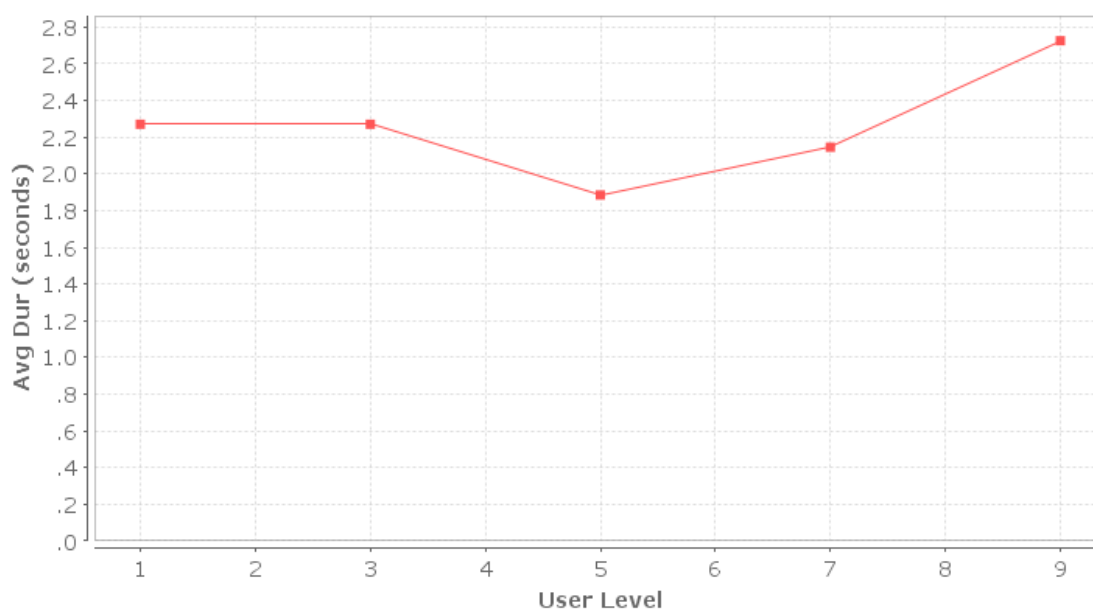
The screenshot shows the configuration interface for a web performance load test. It includes sections for Test Duration, Data Collection, and Virtual Users. The test is set to run for 10 minutes. Data collection is configured with a 10-second sample period, including detailed page durations and individual URL metrics. The virtual users section is configured to start with 2 users, increase by 2 users every 1 minute, and limit to a total of 10 users. A 'Start' button is visible on the right. Below the configuration, a table lists the test cases.

Testcase	Weight	%	Speed	Think Time	VU Start	Delay	Repeats
recording [5]	100	100%	5 Mbps	Recorded	Random	1	n/a

Figura 46. Escenario prueba 1.

Durante 10 minutos se genero una carga de usuarios concurrentes sobre el maestroPQPortlet, esta carga de usuarios inicia en 2 usuarios y es incrementada en 2 cada minuto.

Luego de hacer varias pruebas se encontró que al tener configurado el log⁶⁶ de la aplicación en modo ALL (Guardar todos los mensajes internos en un archivo), los tiempos de respuesta promedios se encontraban en 8 segundos en promedio. Con el fin de aumentar el rendimiento de la aplicación, se optó por configurar el nivel del log en modo INFO (Guardar los mensajes internos de mayor importancia), de esta manera se disminuyó drásticamente el almacenamiento de mensajes en el log de la aplicación, obteniendo como resultado final una reducción de los tiempos de respuesta.



Testcase	Page Name	Avg Dur	Time	Users	Base Dur
• recording [5] (1)	Administración de nodos - SERVIPORTALES	2.726	00:10:00	10	5.001

Figura 47. Tiempos de respuesta 10 usuarios.

Una vez se efectuaron las respectivas modificaciones el tiempo promedio de respuesta paso de 8 segundos a 2.726 segundos en promedio, con una carga máximo de 10 usuarios.

⁶⁶ Utilidad que permite almacenar en un archivo los mensajes internos del sistema para facilitar la trazabilidad y detección de errores

Haciendo uso de la herramienta JMeter, opensource, se realizo la siguiente prueba

Grupo de Hilos

Nombre:

Comentarios

Acción a tomar después de un error de Muestreador

☒ Continuar ☐ Comenzar siguiente iteración ☐ Parar Hilo ☐ Parar Test ☐ Parar test ahora

Propiedades de Hilo

Número de Hilos

Periodo de Subida (en segundos):

Contador del bucle: ☐ Sin fin

☐ Planificador

Figura 48. Configuración JMeter.

Se van a realizar 50 peticiones una después de la otra cada 30 segundos o una vez finalizada las 50 peticiones. Esta prueba será ejecutada 40 veces.

Muestra #	Tiempo de comienzo	Nombre del hilo	Etiqueta	Tiempo de Muestra (ms)	Estado	Bytes
1	22:39:47.163	SERVIPORTALES 1-3	Petición HTTP	1022		39370
2	22:39:48.562	SERVIPORTALES 1-2	Petición HTTP	1630		39370
3	22:39:47.763	SERVIPORTALES 1-4	Petición HTTP	434		39364
4	22:39:45.960	SERVIPORTALES 1-1	Petición HTTP	2240		39365
5	22:39:48.193	SERVIPORTALES 1-2	Petición HTTP	235		39366
6	22:39:48.200	SERVIPORTALES 1-1	Petición HTTP	233		39364
7	22:39:48.186	SERVIPORTALES 1-3	Petición HTTP	282		39368
8	22:39:48.198	SERVIPORTALES 1-4	Petición HTTP	288		39368
9	22:39:48.364	SERVIPORTALES 1-5	Petición HTTP	309		39368
10	22:39:48.429	SERVIPORTALES 1-2	Petición HTTP	272		39370
11	22:39:48.469	SERVIPORTALES 1-3	Petición HTTP	257		39368
12	22:39:48.433	SERVIPORTALES 1-1	Petición HTTP	296		39366
13	22:39:48.486	SERVIPORTALES 1-4	Petición HTTP	257		39368
14	22:39:48.674	SERVIPORTALES 1-5	Petición HTTP	199		39370
15	22:39:48.702	SERVIPORTALES 1-2	Petición HTTP	271		39368
16	22:39:48.729	SERVIPORTALES 1-1	Petición HTTP	257		39370
17	22:39:48.744	SERVIPORTALES 1-4	Petición HTTP	249		39368
18	22:39:48.727	SERVIPORTALES 1-3	Petición HTTP	266		39368
19	22:39:48.874	SERVIPORTALES 1-5	Petición HTTP	226		39368
20	22:39:48.973	SERVIPORTALES 1-2	Petición HTTP	221		39366
21	22:39:48.994	SERVIPORTALES 1-3	Petición HTTP	218		39366
22	22:39:48.987	SERVIPORTALES 1-1	Petición HTTP	233		39366
23	22:39:48.993	SERVIPORTALES 1-4	Petición HTTP	262		39369
24	22:39:48.963	SERVIPORTALES 1-6	Petición HTTP	313		39370
25	22:39:49.101	SERVIPORTALES 1-5	Petición HTTP	177		39368
26	22:39:49.194	SERVIPORTALES 1-2	Petición HTTP	184		39366

☐ Scroll automatically? ☐ Child samples? No. de Muestras 1000 Última Muestra 160 Media 388 Desviación 233

Figura 49. Resultados JMeter.

Luego de realizar esta prueba se encontró que el tiempo promedio que toma hacer cada petición fue de 388 milisegundos con una desviación de 233 milisegundos.

En base a los resultados expuestos anteriormente se encuentra que el sistema maneja los tiempos de respuesta esperados para la carga máxima esperada de usuarios.

Finalizada la ejecución de estas pruebas se detectó e identificó que el uso inadecuado de la herramienta log4j, empleada para imprimir los mensajes internos de la aplicación, puede generar problemas de rendimiento en las aplicaciones.